



João Carlos Moura Évora Rodrigues

Licenciatura em Ciências de Engenharia Biomédica

Segmentação, seguimento e avaliação automática de bactérias em imagens de microscópio

Dissertação para obtenção do Grau de Mestre em
Engenharia Biomédica

Orientador: José Manuel Fonseca, Professor Doutor, FCT/UNL

Co-orientador: André Damas Mora, Professor Doutor, FCT/UNL

Júri:

Presidente: Prof. Doutor Mário António de Basto Forjaz Secca

Arguente: Prof. Doutor Paulo António Martins Ferreira Ribeiro

Vogais: Prof. Doutor José Manuel Matos Ribeiro da Fonseca

Prof. Doutor André Teixeira Bento Damas Mora



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Março 2013



João Carlos Moura Évora Rodrigues

Licenciatura em Ciências de Engenharia Biomédica

**Segmentação, seguimento e avaliação
automática de bactérias em imagens de
microscópio**

Dissertação para obtenção do Grau de Mestre em

Engenharia Biomédica

Orientador: José Manuel Fonseca, Professor Doutor, FCT/UNL

Co-orientador: André Damas Mora, Professor Doutor, FCT/UNL

Júri:

Presidente: Prof. Doutor Mário António de Basto Forjaz Secca

Arguente: Prof. Doutor Paulo António Martins Ferreira Ribeiro

Vogais: Prof. Doutor José Manuel Matos Ribeiro da Fonseca

Prof. Doutor André Teixeira Bento Damas Mora



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Março 2013

Segmentação, seguimento e avaliação automática de bactérias em imagens de microscópio

Copyright© 2013 - Todos os direitos reservados. João Carlos Moura Évora Rodrigues. Faculdade de Ciências e Tecnologia. Universidade Nova de Lisboa. A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

“Os dias prósperos não vêm ao acaso; são granjeados, como as searas, com muita fadiga e com muitos intervalos de desalento.”

Camilo Castelo Branco

Adoro-te mãe.

Agradecimentos

O meu sentido agradecimento aos professores José Manuel Fonseca e André Damas Mora pela orientação, motivação e disponibilidade ao longo dos vários meses. O apoio de ambos foi extremamente importante ao longo de todo o projeto.

O meu especial obrigado à minha colega e amiga Tânia Tomaz, pela paciência e pelo apoio.

Tenho também de agradecer a todos os meus colegas e amigos que tornaram esta importante etapa em algo ainda mais especial. Obrigado pela amizade e companheirismo.

Por último, tenho que agradecer à minha família, nomeadamente aos meus pais e ao meu irmão. Eles são parte importante da minha vida e sem eles nada disto faria sentido.

Resumo

Ao longo de várias décadas, os diversos trabalhos realizados no campo da microscopia caracterizaram-se por um vasto conjunto de procedimentos de análise em imagens microscópicas. A quantificação celular das imagens em estudo é normalmente um procedimento lento, podendo apresentar uma percentagem de erro significativa, devido essencialmente à elevada quantidade de observações a serem efetuadas. Neste sentido, o desenvolvimento de algoritmos de processamento de imagem e de sistemas de reconhecimento, permite a criação de processos de quantificação automática de muitas das imagens microscópicas em estudo.

A dissertação de mestrado aqui apresentada descreve e avalia o desenvolvimento de um algoritmo que tem como objetivo efetuar a segmentação e avaliação de um conjunto de imagens microscópicas. O protótipo projetado constitui um sistema de contabilização automática do número de bactérias *E.coli* visualizado em cada uma das imagens consideradas. As imagens foram disponibilizadas pelo *Laboratory of Biosystem Dynamics* da *Tampere University of Technology* tendo sido adquiridas através de microscópios confocais.

O algoritmo desenvolvido pode dividir-se em três passos distintos. Inicialmente é aplicado um pré-processamento sobre as imagens em estudo constituído por um conjunto de transformações que retiram alguma da informação desnecessária da imagem e ao mesmo tempo melhoram os contornos dos segmentos constituintes. Seguidamente é implementado um processo de *Template matching*, que efetua a detecção da localização de cada uma das bactérias. Neste passo, as bactérias são povoadas individualmente por um conjunto de marcas que são posteriormente utilizadas no terceiro e último passo. Neste último passo é aplicada uma técnica de segmentação baseada no método *Watershed*, um dos mais estudados métodos de segmentação, inseridos na área do processamento de imagem.

O protótipo foi testado nas imagens disponibilizadas, tendo obtido um grau de eficiência bastante satisfatório. De forma a avaliar a versatilidade do software desenvolvido, este foi também aplicado a imagens fornecidas por um dos mais eficientes softwares existentes no mercado e os seus resultados comparados.

Palavras Chave: Segmentação, imagens microscópicas, processamento de imagem, *Template matching*, *Watershed*.

Abstract

Over several decades, published studies in microscopy field were characterized by a wide range of analysis procedures in microscopic images. The quantification of cellular images is usually a slow process, with an important error rate, mainly due to the high number of required observations. Thus, the development of image processing and pattern recognition systems allows the optimization of the automatic quantification processes of many microscopic imaging studies.

This Master's dissertation describes and evaluates the development of a segmentation algorithm, applied to a set of microscopic images. The designed prototype is able to process images sequences for automatic counting of *E.coli* bacteria on each of the images. The set of images was provided by the *Biosystems Dynamics Laboratory* of the *Tampere University of Technology* and was acquired using confocal microscopes.

The algorithm is divided into three distinct steps. In first step a preprocessing set of techniques is applied to the images. These techniques consist on a sequence of image transformations that removes some redundant information and, simultaneously, enhance the contours of each segment. On the second step, a *Template matching* algorithm is computed, resulting on a bacteria location detection procedure. All bacteria are individually populated by a set of marks that are later used by the third and last step. In the final step, it is applied a segmentation technique based on the *Watershed* method, one of the most applied segmentation methods in image processing.

The developed prototype has been tested in the provided images with a satisfactory efficiency. In order to evaluate the versatility of the developed software, it was applied to a different type of images, provided by one of the most efficient segmentation software on the market and the results were compared.

Keywords: Segmentation, microscopic images, images processing, *Template matching*, *Watershed*.

Índice

Índice de Figuras	xv
Índice de Tabelas	xvii
Lista de abreviaturas e siglas	xix
1 Introdução.....	1
1.1 Processamento de Imagem aplicado a imagens microscópicas	1
1.2 Segmentação de imagens.....	2
1.3 Seguimento celular.....	3
1.4 Objetivos propostos.....	3
1.5 Softwares de segmentação e seguimento	4
1.6 Estrutura da dissertação	6
2 Fundamentação Teórica	7
2.1 Métodos gerais de segmentação	7
2.2 Técnicas de pré-processamento	10
2.2.1 Técnicas de alisamento da imagem (<i>Smoothing</i>)	10
2.2.2 Técnicas morfológicas.....	11
2.2.3 Técnicas de evidenciação de contornos	13
2.3 <i>Template matching</i>	14
2.3.1 Transformação de <i>Hough</i> (Fundamentos básicos).....	14
2.3.2 <i>Template matching</i> – algoritmos de correlação	17
2.3.3 <i>Template matching</i> – aplicações gerais	18
2.3.4 <i>Template matching</i> – aplicações em microscopia	20
2.4 <i>Watershed</i>	21
2.4.1 <i>Watershed</i> – Fundamentos básicos.....	22
2.4.2 Aplicações gerais	26
2.4.3 Aplicações em imagens microscópicas	27
3 Metodologia Implementada	31
3.1 Imagens disponibilizadas	32
3.2 Linguagem de programação <i>C#</i>	33
3.3 Biblioteca <i>OpenCV</i>	33
3.4 Pré-processamento	33
3.4.1 Técnica de alisamento da imagem (Filtro Mediana)	33
3.4.2 Técnica de melhoramento de contraste (Transformada de Laplace)	34
3.4.3 Técnica morfológica (Operação de Abertura)	35
3.4.4 Operação de remoção do ruído de fundo	35
3.4.5 Procedimento de redução dos pontos de elevada intensidade presentes nas bactérias	36

3.5 <i>Template matching</i>	40
3.5.1 Implementação geral	42
3.5.2 Definição dos <i>templates</i> gerais	45
3.5.3 Definição de um <i>template</i> específico de detecção das extremidades das bactérias	49
3.5.4 <i>Template matching</i> - Esquematização final	51
3.6 <i>Watershed</i>	53
3.6.1 Aquisição e tratamento das marcas geradas	53
3.6.2 Dilatação das marcas gerais	53
3.6.3 Contagem de marcas	54
3.6.4 Algoritmo <i>Watershed</i>	54
3.7 Seguimento automático.....	58
3.7.1 Processo geral de correção de segmentos.....	58
3.7.2 Processo de seguimento automático	59
3.7.3 Correção do processo de seguimento automático.....	59
4 Resultados obtidos	63
4.1 Aplicação do modelo de segmentação às imagens disponibilizadas	63
4.1.1 Procedimento geral	63
4.1.2 Resultados Obtidos	67
4.1.3 Análise de resultados	70
4.2 Imagens <i>Schnitzcells</i>	71
4.2.1 Procedimento geral	71
4.2.2 Resultados Obtidos	73
4.2.3 Análise de resultados	74
4.3 Seguimento automático.....	75
5 Conclusão.....	77
5.1 Considerações finais	77
5.2 Desenvolvimentos futuros	78
6 Bibliografia	79

Índice de Figuras

Figura 2.1: Distância entre x e y no interior do conjunto A (retirado de [59]).	22
Figura 2.2: Processo de esqueletização num dado conjunto X (retirado de [57]).	23
Figura 2.3: Processo de formação das linhas <i>Watershed</i> (adaptado de [57]).	24
Figura 2.4: Processo de recursão entre X_i e X_{i+1} (adaptado de [55]).	25
Figura 3.1: Exemplos de imagens disponibilizadas. a) Imagem 102 – Sequência 1; b) Imagem 54 – Sequência 2; Imagem 70 – Sequência 5.	32
Figura 3.2: Exemplo de uma aplicação do operador Laplaciano implementado.	34
Figura 3.3: Exemplos da aplicação do procedimento de <i>threshold</i> aqui descrito.	35
Figura 3.4: Exemplo do procedimento de remoção de ruído implementado.	36
Figura 3.5: Dois dos exemplos que ilustram a evidenciação dos pontos de saturação presentes nas imagens.	36
Figura 3.6: Exemplo das consequências da existência dos pontos de saturação após o pré-processamento.	37
Figura 3.7: Aplicação do processo de mediana para a imagem I , tendo em conta as duas imagens anteriores e as duas imagens posteriores.	38
Figura 3.8: Aplicação do processo de mínimo para a imagem I , tendo em conta as duas imagens anteriores e as duas imagens posteriores.	39
Figura 3.9: Esquema resumido do conjunto de procedimentos de Pré-processamento.	40
Figura 3.10: Exemplos de situações de união entre objetos a serem segmentados.	41
Figura 3.11: Exemplo de um mapa de intensidades resultante do processo implementado.	41
Figura 3.12: Esquema ilustrativo do processo de translação do <i>template</i> sobre a imagem.	43
Figura 3.13: Sequência de <i>templates</i> definida.	45
Figura 3.14: Esquema ilustrativo de um dos <i>templates</i> .	46
Figura 3.15: Exemplos das aplicações de cada <i>template</i> sobre uma imagem sintética ($\mu_{template}=0,75$).	47
Figura 3.16: : Exemplos das aplicações de cada <i>template</i> sobre uma imagem real e pré-definida ($\mu_{template}=0,60$).	48
Figura 3.17: Representação dos vários <i>templates</i> aplicados, mediante o processo de rotação implementado.	48
Figura 3.18: <i>Template</i> utilizado no processo aqui descrito.	49
Figura 3.19: Aplicação do procedimento sem ter em conta o processo de detecção de extremidades($\mu_{template}=0,60$).	49
Figura 3.20: Aplicação do procedimento tendo em conta o processo de detecção de extremidades($\mu_{template}=0,60$ e $\mu_{extremidade}=0,68$).	50
Figura 3.21: Exemplo da aplicação do <i>template</i> aqui descrito.	51
Figura 3.22: Fluxograma geral do processo de <i>Template matching</i> implementado.	52
Figura 3.23: Exemplificação do processo de tratamento de marcas aqui descrito.	54
Figura 3.24: Exemplo do resultado da exclusão de segmentos de dimensões reduzidas – a) imagem de entrada; b) imagem segmentada; c) imagem após a aplicação do processo de remoção de segmentos de pequenas dimensões.	55
Figura 3.25: Exemplificação do resultado obtido através do processo de segmentação implementado – a)imagem de entrada; b)imagem resultante do processo <i>Template matching</i> com as marcas correspondentes; c)processo de segmentação através do método <i>Watershed</i> .	55
Figura 3.26: Exemplificação do procedimento de aquisição automática de um ficheiro contendo o número de bactérias presente nas imagens em estudo.	56
Figura 3.27: Fluxograma do procedimento <i>Watershed</i> implementado neste projeto.	57
Figura 3.28: Ilustração exemplificativa de uma incorreção entre <i>frames</i> consecutivos.	58

Figura 3.29: Exemplo real de um procedimento de correcção do seguimento (sequência de imagens <i>Schnitzcells</i>).....	61
Figura 4.1: Sequência II-Imagem 38.....	63
Figura 4.2: Sequência I-Imagem 60.....	63
Figura 4.3: Sequência IV-Imagem 72.....	64
Figura 4.4: Sequência III-Imagem 45.....	64
Figura 4.5: Sequência V-Imagem 24.....	64
Figura 4.6: Fluxograma relativo aos passos gerais do algoritmo implementado.....	65
Figura 4.7: Resultados do algoritmo de segmentação implementado quando aplicado às imagens ilustradas nas Figuras 4.1 a 4.5.	66
Figura 4.8: Ilustração exemplificativa da interface gráfica criada de forma a auxiliar o processo de contagem manual.....	67
Figura 4.9: Comparação da taxa de detecção média para o conjunto das 25 imagens analisadas.	69
Figura 4.10: Algumas das imagens “ <i>Schnitzcells</i> ” e respetivo resultado de segmentação através do software “ <i>Schnitzcells</i> ”.	72
Figura 4.11: Algumas imagens resultantes da segmentação através do software desenvolvido neste projeto.....	72
Figura 4.12: Gráfico ilustrativo do número de bactérias detectado.	73
Figura 4.13: Comparação da taxa de detecção média para o conjunto das 20 imagens analisadas.	74

Índice de Tabelas

Tabela 4.1: Número de bactérias presentes em cada imagem analisada.....	68
Tabela 4.2: Módulo das diferenças entre o número de bactérias reais e o número detectado automaticamente.....	68
Tabela 4.3: Taxas de detecção entre o número real de bactérias e o número detectado automaticamente.....	69
Tabela 4.4: Avaliação de critérios de eficiência do modelo de segmentação.....	70
Tabela 4.5: Número de bactérias presente em cada uma das 20 imagens “ <i>Schnitzcells</i> ”	73
Tabela 4.6: Módulo da diferença entre o número de bactérias real e o número de bactérias detectado, para cada algoritmo.....	73
Tabela 4.7: Avaliação de critérios de eficiência do modelo de segmentação.....	73
Tabela 4.8: Percentagens de erro entre o número real de bactérias e o número detectado automaticamente.....	74

Lista de abreviaturas e siglas

E.coli *Escherichia coli*

B.subtilis *Bacillus subtilis*

SKIZ *Skeleton by influence zones*

CHiTA *Circular Hough image transform algorithm*

SIFT *Scale Invariant Feature Transform*

Ciratefi *Circular, Radial and Template-Matching Filter*

THG *Third Harmonic Generation*

CLI *Common Language Infrastructure*

CTS *Common Type System*

MLL *Machine Learning Library*

MD Módulo das diferenças

ND Número de bactérias detectado

NR Número real de bactérias

TD Taxa de detecção

1 Introdução

1.1 Processamento de Imagem aplicado a imagens microscópicas

O processamento de imagem é uma das áreas tecnológicas de maior crescimento nas últimas décadas. Os desenvolvimentos realizados nesta área têm por base a importância inerente da tecnologia associada à imagem digital [1]. A imagem digital, assim como todos os processos associados aos sistemas digitais, são cada vez mais aplicados em diversas vertentes. Entre os possíveis exemplos, destacam-se, entre outros, as aplicações em sistemas sensoriais, fotografia e monitorização. Os constantes avanços tecnológicos têm permitido a extração automática de toda a informação necessária relativamente à imagem. Este automatismo é cada vez mais importante neste campo, tornando possível a criação de modelos cada vez mais rápidos e eficientes.

Muitos dos desenvolvimentos introduzidos no processamento de sistemas digitais estão relacionados com o ramo da medicina, mais precisamente com a análise de imagens médicas. Os constantes avanços nesta área permitem uma aplicação mais prática e de importância vital na resolução de problemas médicos. O estudo de imagens microscópicas assume-se como parte integrante desta área. Através da análise de um conjunto de amostras biológicas são analisadas características específicas (forma; cor; dimensão; textura...) dos objetos previamente identificados, que podem ser úteis na indicação de diagnósticos patológicos mais precisos e eficazes [2]. Tendo isto em consideração, o estudo de microrganismos apresenta um elevado impacto económico na saúde humana, assim como noutras áreas como, por exemplo, nos recursos ambientais.

Na sua generalidade as imagens microscópicas apresentam vários tipos de microrganismos que podem apresentar-se de forma individual ou formar colónias que podem conter centenas ou milhares de células. Existem quatro tipos principais de microrganismos: bactérias, *fungi*, *protozoos* e vírus [3]. O processo manual de validação de bactérias em visualizações microscópicas exige uma observação cuidada e muito exigente, tendo em conta a complexidade deste tipo de imagens e em muitos casos, um elevado número de visualizações. Este processo é, na maioria das vezes, lento e apresenta uma significativa percentagem de erro. O largo conjunto de técnicas desenvolvido nas últimas décadas na área do processamento de imagem permite a extração de informação adicional à validação e caracterização manual. Com o objetivo de quantificar automaticamente as amostras, têm sido desenvolvidos vários algoritmos baseados nas técnicas existentes de processamento de imagem e nas técnicas baseadas no reconhecimento automático de padrões. Neste contexto, uma das mais importantes áreas inseridas na análise automática de imagens é a denominada segmentação de imagem. A segmentação de imagem permite, entre outras coisas, uma melhor extração da informação presente numa imagem em estudo, tornando possível uma melhor compreensão da mesma.

1.2 Segmentação de imagens

A ideia geral da segmentação é inicialmente separar o fundo das imagens de tudo o resto e seguidamente separar a imagem em diferentes regiões de interesse que definem cada objeto presente nesta. Este método é utilizado para simplificar a imagem e ao mesmo tempo realçar as suas características principais. Tendo isto em consideração, este método apresenta uma grande importância na análise de imagens, pois faz parte de um passo indispensável para que mais tarde seja possível identificar os objetos, extrair as características necessárias e proceder às devidas classificações. Desenvolver um algoritmo que efetue a segmentação com sucesso torna-se fulcral para o correto funcionamento de todo o processo. A metodologia de segmentação depende muito das características da imagem e do tipo de objetos de interesse [1].

Como descrito em [4], podem-se considerar três propriedades básicas que facilmente influenciam o tipo de abordagem a utilizar no processo de segmentação.

- **Cor:** Esta propriedade assume grande importância quando os objetos a serem identificados apresentam cores distintas relativamente à cor de fundo, sendo facilmente separados deste mesmo fundo.
- **Textura:** Esta propriedade encaixa nas situações em que os objetos apresentam variações espaciais de intensidade dentro de uma determinada escala. É então feita uma medição baseada no cálculo da variância ou qualquer outro parâmetro estatístico relativo à intensidade dos pixels da imagem.
- **Movimento:** Esta propriedade é tida em consideração quando se trabalha com sequências de imagens (*frames*). Quando os objetos se sobrepõem a um fundo estacionário, uma simples subtração *frame* por *frame* é muitas vezes utilizada para proceder à segmentação.

O tipo de abordagem a utilizar para proceder à segmentação tem geralmente em conta uma combinação destas e de outras propriedades. Tendo em conta os desenvolvimentos recentes nesta área, é tipicamente direcionada para uma de duas vertentes [4]:

- **Edge/boundary methods:** Este método baseia-se na identificação de contornos de forma a detectar os limites de cada objeto presente na imagem. Como tal, tem em conta as diferenças de forma entre os diferentes grupos de pixels.
- **Region-based methods:** Este método baseia-se na designação dos pixels de uma determinada região, consoante o grau de similaridade.

O primeiro passo para analisar a imagem é o chamado pré-processamento. O pré-processamento é necessário para melhorar a qualidade da imagem, eliminando ruído e todo o tipo de artefatos que prejudiquem a sua análise. É muito importante garantir que este pré-processamento não remova informação significativa relativa aos objetos presentes na imagem em estudo. Algumas das técnicas mais importantes de pré-processamento são descritas no **capítulo 2.2**.

1.3 Seguimento celular

A avaliação de várias características biológicas está muitas vezes interligada com o entendimento do comportamento celular. Os processos dinâmicos associados ao comportamento celular podem ser adquiridos através de vários processos digitais. Uma das principais classes inseridas no ramo do processamento de imagem está intimamente relacionada com o seguimento temporal e automático de determinadas amostras celulares. O enorme crescimento do conjunto de tecnologias associadas a imagens microscópicas tem como consequência uma série de desafios cada vez mais exigentes, no que diz respeito ao desenvolvimento de algoritmos de seguimento automático [5]. Seguidamente será feita uma breve abordagem relativa a alguns dos desenvolvimentos inseridos neste campo.

O seguimento automático de leucócitos foi introduzido por Acton et al., sendo apresentada uma técnica que executa o seguimento automático de leucócitos [6]. Esta técnica é baseada na identificação de contornos, nomeadamente no processamento de curvas paramétricas ou não paramétricas, que se movem sobre o plano da imagem e capturam os limites dos objetos considerados. Estas curvas foram já frequentemente utilizadas em aplicações de *Tracking*, sendo denominadas de *Snake curves*. A técnica introduzida por Acton et al. é comparada com algoritmos de *Tracking* baseados em sistemas de correlação ou de extração de características morfológicas e apresenta a capacidade de processar as amostras de leucócitos com um maior grau de eficiência, quando comparado com outras técnicas.

Um dos principais problemas encontrados pelos algoritmos de seguimento automático é a existência de fenómenos de subssegmentação e sobressegmentação, que claramente afetam os resultados obtidos. Mosig et al. propuseram um modelo baseado em alinhamentos topológicos, de forma a fazer face a este tipo de problemas [5]. É feita uma análise relativa a cada um dos vários *frames* disponibilizados em várias amostras, sendo criado um sistema que identifica os vários momentos de divisão celular entre *frames* consecutivos. Wang et al. apresentaram um método automático de reconstrução temporal em imagens microscópicas [7]. Este método combina um conjunto de técnicas morfológicas de segmentação, assim como um método de contagem de valores baseado numa correspondência entre os *frames* consecutivos. Este modelo foi aplicado a vários tipos de imagens microscópicas, como amostras de bactérias ou células humanas, apresentando uma grande portabilidade e eficiência.

1.4 Objetivos propostos

Nesta dissertação é descrita uma aplicação que foi desenvolvida com o sentido de proceder ao seguimento e avaliação de bactérias em imagens de microscópio confocal, adquiridas no laboratório (*Laboratory of Biosystem Dynamics da Tampere University of Technology*). A aplicação desenvolvida é apresentada sob a forma de um protótipo de software que permite efetuar o processamento de sequências de imagens microscópicas. Ao longo deste projeto foram consolidadas diversas técnicas de processamento digital que em conjunto permitem identificar e caracterizar sequências de imagens contendo colónias de bactérias. As técnicas desenvolvidas foram implementadas na linguagem C#, com recurso à biblioteca do *OpenCV*.

O processo de identificação das bactérias contidas nas imagens é dividido em três passos principais. Inicialmente são aplicadas técnicas de pré-processamento que têm como principal objetivo melhorar a qualidade da imagem, no sentido de evidenciar os contornos individuais das várias bactérias que se agrupam em pequenas colónias e de retirar parte da informação desnecessária da imagem, como o ruído ou algumas imperfeições contidas no fundo da imagem. Estas técnicas são detalhadamente descritas no **capítulo 2.2**. Dentro das técnicas aplicadas destaca-se a aplicação de um operador de Laplace que tem como objetivo a detecção das bordas dos objetos contidos nas imagens. Os fundamentos básicos deste operador são descritos na **secção 2.2.3**.

Seguidamente ao pré-processamento é aplicado um método de *Template matching* baseado num algoritmo de correlação. Este método é aplicado com o intuito de comparar a imagem em estudo com um *template* pré-definido, de forma a detectar objetos na imagem com relativa semelhança a este *template*, identificando assim a localização de cada bactéria contida na imagem. É aplicada uma sequência definida de *templates* que tem em conta a forma e o tamanho das bactérias que se espera estarem presentes na imagem em estudo. Os fundamentos básicos relativos ao *Template matching* são detalhadamente descritos no **capítulo 2.3**. As especificações do modelo implementado neste projeto são introduzidas no **capítulo 3.5**.

O resultado proveniente da aplicação do método *Template matching* é a visualização de pequenas marcas que localizam as diferentes bactérias contidas na imagem. Estas marcas são posteriormente utilizadas num processo de *Watershed* com marcas. Este é um dos métodos de segmentação mais utilizados e que permite a segmentação da imagem e a contagem de bactérias. Os fundamentos básicos relativos a este método são descritos no **capítulo 2.4**. As especificações do modelo implementado neste projeto são descritas no **capítulo 3.6**.

Após o processo de identificação e contagem das bactérias existentes nas imagens em estudo, é realizado um processo de seguimento automático que permite a observação temporal de sequências de amostras microscópicas, sendo possível extrair várias propriedades, como a mobilidade celular, a velocidade de reprodução ou a origem temporal de cada célula (**capítulo 3.7**).

1.5 Softwares de segmentação e seguimento

Seguidamente será feita uma abordagem relativa a alguns dos softwares de segmentação e de seguimento em imagens celulares. Os sistemas aqui descritos fazem parte de um conjunto de softwares disponíveis gratuitamente que foram avaliados ao longo deste trabalho.

CellTrack:

Este software foi publicado em 2008 e apresenta-se como um *Open-Source* que permite a análise automática do seguimento celular, avaliando também a mobilidade temporal das amostras celulares [8]. Esta característica celular apresenta-se como uma parte muito importante no estudo de vários processos biológicos. Neste sentido, é apresentado um modelo de avaliação celular baseado nas propriedades morfológicas das partículas em estudo. O software contém vários métodos que executam alguns passos chave, englobando o pré-processamento, a segmentação e o seguimento automático. É também incluído um sistema de

seguimento dos vários segmentos a partir dos contornos correspondentes. Este sistema apresenta um elevado grau de eficiência, mesmo em situações de deformação celular.

CellTracer:

O software *CellTracer* é uma aplicação que permite a utilização de algumas ferramentas automáticas de segmentação e reconstrução temporal [9]. A aplicação é baseada nas ferramentas de processamento de imagem fornecidas pelo Matlab e permite a utilização de alguns procedimentos manuais ou semi-automáticos de correção de alguns problemas resultantes do processo automático. A eficiência da aplicação é testada através da aplicação em vários tipos de conjuntos celulares. O processo de segmentação e seguimento automático é dividido em diferentes passos, passando por uma extração da região de fundo da imagem e seguidamente por um processamento das regiões dos contornos existentes.

CellTracker:

O software *CellTracker* permite a execução do seguimento nuclear e citoplasmático, através dos picos de intensidade fluorescente e da análise dinâmica de alguns fatores transcritivos [10]. Este software foi projetado para processos de avaliação em células eucariontes e tem vindo a ser desenvolvido desde 2009 pelo *Warwick Systems Biology Centre*.

TLM-Tracker:

O *TLM-Tracker (Time-Lapse-movie-Tracker)* é uma ferramenta de avaliação de imagens celulares que permite a identificação automática de células individuais e o seguimento de sequências temporais [11]. Esta plataforma foi testada em colónias de *Bacillus megaterium*. Adicionalmente, permite a quantificação de marcas específicas de fluorescência e a determinação de algumas características individuais. O software requer a análise de imagens de fluorescência de elevada intensidade.

Schnitzcells:

O software *Schnitzcells* executa a segmentação e seguimento de imagens celulares contendo células procariontes, nomeadamente, *E.coli* e *B.subtilis* [12]. Este software permite a análise de sequências temporais de imagens de fluorescência ou de contraste de fase. Para além dos processos automáticos de segmentação, foi incluído um sistema que permite ao utilizador editar a informação das imagens resultantes, de forma a corrigir determinados erros de segmentação, nomeadamente, problemas de sobressegmentação. O modelo foi desenvolvido em Matlab, fornecendo uma interface gráfica bastante intuitiva que pode ser utilizada pela generalidade dos utilizadores. Toda a informação extraída pode ser guardada numa única estrutura, que pode posteriormente ser utilizada, tanto para uma simples visualização, como para procedimentos de avaliação adicionais. O software apresenta um grau de flexibilidade elevado, podendo ser adaptado a várias condições experimentais.

Tendo em conta as propriedades do software, assim como o grau de versatilidade e eficiência reconhecido, este produto é utilizado como referência neste trabalho, sendo estabelecido um procedimento de comparação entre os resultados obtidos pelo software *Schnitzcells* e o software aqui desenvolvido. Este procedimento permite avaliar a eficiência da aplicação agora criada sendo apresentado no **capítulo 4.2**.

1.6 Estrutura da dissertação

A presente dissertação está organizada em alguns pontos principais. Seguidamente é feita uma abordagem teórica relativa aos fundamentos que servem de base a todas as técnicas inseridas no protótipo desenvolvido (**capítulo 2**). Adicionalmente são descritos alguns estudos já realizados que foram baseados nestes mesmos fundamentos. O capítulo é dividido nos três passos principais do projeto desenvolvido: 1- Pré-processamento; 2- *Template matching* e 3- *Watershed*.

O **capítulo 3** descreve pormenorizadamente cada uma das técnicas implementadas, sendo os resultados obtidos através da aplicação do conjunto de técnicas e apresentados passo a passo.

Todos os procedimentos de avaliação da eficiência do algoritmo desenvolvido são apresentados no **capítulo 4**. Este capítulo está dividido em três secções. A primeira secção avalia a aplicação do algoritmo sobre as imagens disponibilizadas. Na segunda secção são apresentados os resultados obtidos através de um processo de comparação entre o protótipo desenvolvido e um outro software já existente. Na terceira e última são descritos os resultados referentes ao processo de seguimento automático. No **capítulo 5** é efetuada a avaliação dos resultados, sendo feita uma abordagem relativa à eficiência do algoritmo proposto e aos possíveis desenvolvimentos futuros.

2 Fundamentação Teórica

2.1 Métodos gerais de segmentação

Neste capítulo será feita uma abordagem relativa aos métodos gerais de segmentação, aplicados ao longo das últimas décadas em imagens microscópicas de amostras constituídas por colónias de células e que foram tidos em conta neste projeto. A caracterização destas imagens tem-se tornado fundamental no ramo da biologia, respondendo a várias questões que se prendem com a biodiversidade, população genética ou até mesmo com a determinação das características de um determinado sistema biológico (ruído; envelhecimento;...) [13]. A identificação e caracterização destas imagens tem sido, portanto, um desafio, tendo mesmo em conta que a observação de bactérias, sob condições naturais, é na maioria das vezes imprecisa. Esta imprecisão tem a ver com o facto de as células entre si serem de difícil distinção, pois têm tendência a agregarem-se ou até mesmo a sobreporem-se, formando colónias que podem conter centenas ou milhares de células. Neste sentido têm sido desenvolvidos modelos estatísticos e matemáticos que visam à optimização dos métodos produzidos [7].

A grande maioria dos algoritmos de segmentação implementados nas últimas décadas são baseados nos métodos de *Watershed* e em aproximações de *level-set*. O método *Watershed* é descrito no **capítulo 2.4**. O método *level-set* foi primeiramente apresentado por Osher e Sethian em 1987. Neste método são inicialmente definidos os contornos da imagem através de uma determinada função, denominada de “*level-set function*”. O nível zero desta função corresponde à superfície definida pelo contorno de cada objeto. O comportamento desta função depende da equação de Hamilton-Jacobi [14]: $\frac{\partial f}{\partial t} = v|\nabla f|$, em que f é a função level set, v a velocidade e t o tempo. No entanto, a solução numérica desta equação requer técnicas relativamente complexas. Este método é descrito de forma aprofundada em [15]. A localização de marcas (sementes) ou uma pré-identificação celular são fatores a ter em consideração na utilização destes métodos. A portabilidade destes métodos é então claramente dependente do tipo de imagem a utilizar e principalmente do tipo de célula em estudo. Estes modelos “base” têm sido e modificados ao longo do tempo e sofrido constante evolução.

Diversos algoritmos de segmentação foram avaliados nos últimos anos, devido à grande variedade das características das imagens em estudo. Segundo [16], as técnicas de segmentação podem ser categorizadas em três classes:

- 1- Características específicas de *threshold* ou *clustering*;
- 2- Detecção de bordas;
- 3- Extração de regiões específicas.

Na área da segmentação de imagens relacionadas com a biomedicina, muitas das técnicas propostas são direcionadas para as classes 1 e 2. Tendo em consideração que muitos dos métodos já desenvolvidos poderão obter resultados satisfatórios apenas num conjunto muito restrito de imagens [7], um dos grandes objetivos dos esforços atuais prende-se com a necessidade de produzir ferramentas capazes de extrair características de forma independente do tipo de imagem a analisar.

Nas últimas décadas foram desenvolvidos vários métodos com o objetivo de proceder à segmentação de imagens microscópicas que são dependentes das definições dos extremos de intensidade dos pixels da imagem. A aplicação de níveis de *threshold* permite a distinção entre os objetos (de elevada intensidade) e a região do fundo (de baixa intensidade) ou vice-versa [17]. Este tipo de aplicações, na maioria dos casos, não consegue reproduzir os melhores resultados, sendo muitas vezes necessária uma definição manual do nível de *threshold* utilizado. Neste sentido, vários algoritmos têm sido desenvolvidos com o objetivo de conseguirem efetuar a determinação automática dos níveis de *threshold*. Tobias et al. propuseram um método de segmentação baseado num modelo alternativo de determinação do *threshold* que tem em conta a similaridade entre níveis de cinzento [18]. Este modelo apresenta um maior grau de eficiência, quando comparado com outros modelos que são obtidos através da minimização de uma função específica, dependente do *threshold* utilizado. Wilkinson et al. apresentaram uma técnica de segmentação de rápida resolução. Esta técnica é baseada na implementação de um sistema de determinação de diferentes *thresholds* que são aplicados a imagens monocromáticas. Este modelo pode ter aplicações em várias áreas [19]. Forero et al. propuseram um modelo inovador que utiliza um sistema de *thresholds* simples em canais cromáticos e também combina este processo com um sistema de detecção de características relativas à forma dos segmentos. Neste processo é aplicada uma técnica de *k-means clustering* de forma a proceder à respetiva classificação. Um dos principais objetivos da projeção deste modelo é o de reduzir o tempo de análise de algumas amostras de saliva [20]. Russel et al. desenvolveram um sistema estável de contagem dos níveis de *threshold*, de forma a segmentar pequenos núcleos em imagens microscópicas, que apresentam um sistema de digitalização constituído por lasers confocais. Este algoritmo apresenta resultados muito satisfatórios, quando comparado com outros algoritmos de segmentação que utilizam sistemas básicos de *threshold* e apresenta resultados semelhantes aos reproduzidos através de um processo de definição manual de *threshold* [21]. A segmentação de imagem baseada em *thresholds* não apresenta bons resultados nos casos em que existe uma grande variação nas intensidades dos pixels que constituem os objetos. Nestes casos, as técnicas de segmentação baseadas na detecção de contornos funcionam melhor. Este tipo de segmentação identifica segmentos através da procura de algumas descontinuidades, como as bordas de um determinado objeto presente numa imagem. Em alguns casos, o mesmo modelo aplica os dois tipos de segmentação aqui enunciados, como é o caso do modelo desenvolvido por Forero (já aqui enunciado) [20].

Existem vários exemplos de algoritmos baseados na detecção de contornos, aplicados a imagens microscópicas. Tendo em conta o facto da maioria das imagens em estudo apresentarem um nível considerável de desfocagem, torna-se usual a representação do gradiente da imagem, de forma a evidenciar as regiões dos contornos [22]. Muitos dos problemas de segmentação são, na sua forma fundamental, problemas relacionados com a detecção de contornos, no sentido em que a localização dos contornos determina as características de cada célula presente na imagem em consideração. O processo de detecção

de contornos tem por objetivo simplificar a análise das imagens, reduzindo drasticamente a quantidade de informação a ser processada mas preservando a informação útil relativa aos limites de cada objeto [23]. Infelizmente este tipo de técnicas falha nos casos em que se trabalha com imagens que apresentam características muito complexas, com regiões demasiado uniformes, ou contornos muito pouco distintos [24]. Neste sentido, ao longo dos anos, houve uma tendência para desenvolver diversas aplicações que apresentam algumas nuances, de forma a produzirem resultados mais satisfatórios. Ji et al. apresentaram um método que segmenta imagens constituídas por múltiplos canais, utilizando um filtro *Gabor* com um sistema de etiquetagem de contornos [25]. O modelo desenvolvido apresenta melhores resultados no processo de detecção de alguns contornos de baixa intensidade, quando comparado com os algoritmos de detecção de contornos baseados no gradiente da imagem. Shapov desenvolveu um método de implementação de um filtro Laplaciano, dependente de um parâmetro específico de intensidade de cada contorno, reproduzindo uma representação relativamente precisa dos limites de cada célula presente nas imagens em estudo [26].

Geralmente as células presentes nas imagens possuem algumas características específicas, relativas às suas formas geométricas. Estas características têm sido exploradas, no sentido de desenvolver várias soluções de análise. Vallaton et al. apresentaram um algoritmo capaz de identificar partículas em forma de bastonete (tal como bactérias) em imagens *phase contrast* de alta resolução [13]. Este método é baseado num modelo probabilístico, associado a um conjunto padrão de características. Vallaton apresentou também um método capaz de detectar regiões padrão (*ridge-like patterns*), em determinadas imagens *phase contrast*, através da aquisição da intensidade dos pixels da imagem em múltiplas direções e em curtos segmentos de linha. Este método é denominado *linear feature detector* [27]. Hates et al. desenvolveram um sistema de contagem que detecta o número de colónias de bactérias. Este sistema é baseado em algoritmos de segmentação e classificação que têm em conta a forma dos objetos visualizados [28]. O modelo proposto é dividido em duas partes: detecção de padrões (*pattern detector*) e um sistema de contagem. O classificador de padrões testa em que situações uma determinada forma geométrica binarizada corresponde a uma colónia ou a um grupo, de acordo com determinados parâmetros relativos à circularidade do objeto em causa. Grande parte dos métodos utilizados em segmentação de imagem, baseados nas características relativas à forma, utilizam a transformada de *Hough* (este método é descrito na **secção 2.3.1**). Um dos vários exemplos de uma aplicação da transformada de *Hough* em imagens microscópicas é apresentada por Bewes em [29]. Yao et al. desenvolveu um método completamente alternativo, aplicado a imagens contendo partículas semelhantes a grãos de arroz e que pode ser utilizado em várias aplicações clínicas [30]. Este método é baseado na detecção de pontos que representam as extremidades côncavas dos grãos de arroz que se encontram ligeiramente unidos. Estes pontos são localizados através de cálculos relativos à minimização da distancia Euclideana entre pontos específicos.

Veropoulos et al. desenvolveram um método automático de detecção de células, relacionadas com tuberculose, em amostras clínicas. Este método é baseado em técnicas de processamento de imagem e redes neuronais. O algoritmo implementado mostra uma sensibilidade de 93,5% na identificação de células individuais [31]. Wang et al. desenvolveram um conceito inovador em imagens, através de um processo denominado de *gray-scale/black-white*, aplicando um elevado conjunto de operadores matemáticos e permitindo a extração de células de uma determinada imagem, através de um processo que gradualmente converte a imagem *gray-scale* numa imagem *black-white* [7]. Nakib et al. propuseram um algoritmo de segmentação baseado na optimização de um modelo híbrido de anelamento microcanónico. O processo de entropia exponencial (a duas dimensões) aqui considerado tem em conta a

distribuição da informação relativa à intensidade dos pixels e ao mesmo tempo, tira partido da informação espacial, através da utilização de histogramas específicos [32].

2.2 Técnicas de pré-processamento

Ao longo deste capítulo serão descritas algumas técnicas de pré-processamento que foram aplicadas neste projeto [33]. Muitas destas operações aplicam o conceito de vizinhança local de forma a definir a área local de influência dos pixels vizinhos. Neste sentido, torna-se importante introduzir alguns pontos essenciais, relativamente a filtros espaciais e definições de *kernel* [4][2].

Num filtro linear, o valor de determinado pixel é determinado através de algumas combinações lineares entre os pixels vizinhos. A combinação linear envolvida entre os pixels vizinhos é determinada por um *kernel*. Um *kernel* é um pequeno quadrado ou disco que apresenta um ponto “âncora” no centro. O processo de filtragem é implementado através de um sucessivo posicionamento por parte do *kernel*, de forma a que a localização do seu centro coincida com a localização de cada pixel alvo. O *kernel* percorre então toda a região de interesse da imagem a ser processada, multiplicando os pixels, sobrepostos pelo *kernel*, pelos respetivos pesos. Seguidamente é feito o somatório dos valores calculados e o valor resultante é copiado para uma nova imagem, na localização correspondente. Este tipo de computação é expressa por um processo chamado de convolução, sendo que muitos dos *kernels* são várias vezes denominados de *kernels* de convolução.

Os filtros não lineares apresentam o mesmo mecanismo básico que os lineares, mas os valores resultantes são calculados através de uma operação não linear entre os pixels vizinhos.

2.2.1 Técnicas de alisamento da imagem (*Smoothing*)

Este tipo de processamento é usualmente realizado para reduzir o nível de ruído de uma imagem, podendo ser também utilizado para reduzir o número de artefatos ou mesmo para reduzir a resolução da imagem [33].

Filtro de desfocagem (Média): Esta operação pode ser considerada o caso mais simples, em que o valor de cada pixel da imagem resultante é obtido através da média simples de todos os pixels incluídos no *kernel* à volta do pixel correspondente na imagem de entrada. Esta operação é frequentemente realizada para redução de ruído. Quando o *kernel* se situa sobre uma região de elevado contraste provoca obviamente alguma desfocagem e alguma degradação nos contornos, o que pode constituir um grave problema nos casos em que estes representem parte da informação útil da imagem.

Filtro Gaussiano: O operador Gaussiano é um dos operadores mais utilizados em softwares gráficos para reduzir ruído. Considerando um determinado vetor a partir de uma imagem de entrada, é feita a convolução entre cada ponto deste vetor e um *kernel* Gaussiano. Os valores resultantes são depois somados de forma a produzir o valor final.

Filtro mediana: Esta operação é uma das mais utilizadas em processamento de imagem. Cada pixel da imagem resultante é definido como sendo o valor mediano entre os pixels da imagem de entrada que estão sobrepostos pelo *kernel*. O filtro de mediana é

geralmente utilizado para realçar os contornos de uma imagem, pois este filtro tem a capacidade de preservar os contornos e ao mesmo tempo remover parte considerável do ruído da imagem. Para determinados tipos de ruído, este filtro apresenta melhores resultados do que o filtro Gaussiano (anteriormente descrito).

Filtro bilateral: Este filtro faz parte de uma ampla classe de operadores denominados de *edge-preserving smoothing*. O filtro Gaussiano pode não apresentar bons resultados em zonas de elevado contraste, zonas em que é expectável que não haja correlação entre os pixels e os respetivos vizinhos. O filtro bilateral permite o alisamento da imagem e ao mesmo tempo a preservação de zonas de elevado contraste, à custa de um maior tempo de execução. Este operador define uma média ponderada entre cada pixel e os pixels vizinhos, existindo uma componente principal que é baseada na variação de intensidades dos pixels, a partir do pixel central. Quanto maior for este parâmetro, mais ampla é a escala de intensidades incluída no processo de alisamento. A aplicação deste filtro sobre a imagem faz com que esta seja transformada em algo similar a uma pintura em aquarela, o que muitas vezes torna mais fácil um posterior processo de segmentação.

2.2.2 Técnicas morfológicas

A maior parte das transformações morfológicas é utilizada de forma a isolar objetos individuais, juntar segmentos que pertencem ao mesmo objeto ou remover parte considerável do ruído presente na imagem. Estas transformações são também utilizadas em processos de obtenção de gradientes. Os operadores mais básicos apresentam-se como combinações entre processos de erosão e dilatação. Os operadores de erosão e dilatação são utilizados para eliminar ruído ou mesmo alguns contornos excessivamente irregulares [1]. Aqui são descritos alguns destes operadores. Algumas das definições matemáticas podem ser encontradas em [34].

Dilatação: Esta transformação resulta da interseção da imagem com um *kernel* específico. O tamanho e a forma deste *kernel* são definidos pelo utilizador. Numa imagem em escala de cinzentos, o efeito do *kernel* sobre a imagem é semelhante a um operador de máximos locais. De entre os pixels sobrepostos pelo *kernel*, é determinado o pixel com o valor máximo e o valor do pixel central é substituído. O efeito deste operador é o de realçar os pontos de maior intensidade, fazendo com que a visualização dos mesmos seja alargada. Numa imagem binarizada, a imagem resultante é constituída por todos os pixels da imagem original para os quais a interseção com o *kernel* (estando o pixel correspondente sobre o ponto “âncora”) resulte num conjunto não nulo.

O operador de dilatação é frequentemente utilizado quando uma dada região está “partida” em múltiplos segmentos. A ideia geral é unir determinadas regiões com intensidades semelhantes.

Erosão: Nesta operação, o *kernel* desloca-se ao longo da imagem e de entre os pixels sobrepostos pelo *kernel*, é determinado o pixel com o valor mínimo e o valor do pixel central é substituído. O efeito deste operador é o de escurecer os pontos de maior intensidade, fazendo com que a visualização dos mesmos seja diminuída. Numa imagem binarizada, apenas são considerados os pixels nos quais a interseção entre a imagem e o *kernel* (com o ponto “âncora”) sobre o pixel em consideração seja o próprio *kernel*.

Esta operação é frequentemente utilizada para eliminar parte do ruído presente na imagem. Neste caso, o objetivo é erodir pequenos segmentos até desaparecerem e ao mesmo tempo, preservar o conteúdo significativo da imagem.

Operador de abertura e Operador de fecho: Estes operadores são processados através de combinações entre erosões e dilatações. O operador de abertura é muito utilizado em processos de contagem de regiões. Neste caso, geralmente é processado um processo de erosão, seguido por um processo de dilatação.

No operador de fecho é geralmente efetuado um processo de dilatação, seguido por uma erosão. Este operador é frequentemente utilizado para “preencher” segmentos ou para eliminar algum do ruído presente na imagem, sendo muito útil em algoritmos de componentes ligados.

Gradiente morfológico: O objetivo deste operador é o de isolar os contornos dos segmentos contidos numa determinada imagem. Quando este operador é aplicado, o perímetro de uma dada região pode ser determinado, através da subtração da parte expandida em relação à parte interior da região, sendo criado um contorno bem definido. Este operador pode ser representado através da seguinte fórmula:

$$\text{Gradiente}(I) = \text{dilatação}(I) - \text{erosão}(I)$$

Em que I representa a imagem em consideração.

Operador Top Hat e Black Hat: Estes operadores são muito utilizados para isolar regiões com grandes variações de intensidade relativamente às regiões vizinhas. *Top Hat* é usado para isolar regiões de elevada intensidade e *Black Hat* regiões de baixa intensidade. Estes operadores são muito utilizados no processamento de imagens microscópicas e podem ser definidos da seguinte forma.

$$\text{TopHat}(I) = I - \text{Abertura}(I)$$

$$\text{BlackHat}(I) = \text{Fecho}(I) - I$$

Threshold: O processo de *threshold* tem sido desde sempre uma das principais técnicas utilizadas no campo do processamento de imagem. O processo simples de definição de um limiar que determina a seleção ou a rejeição de determinados pixels, consoante o valor dos mesmos, é utilizado há várias décadas em modelos de processamento. No caso de um processo de binarização (o caso mais simples), os valores abaixo do *threshold* são substituídos por 0 e os restantes são substituídos por 1. Para além deste, existem outros processos, tal como a binarização invertida ou processos de truncação, em que os valores abaixo ou acima do *threshold* são substituídos pelo próprio.

Usualmente, o cálculo automático do *threshold* é inicialmente feito através da determinação do histograma da imagem. Um histograma pode ser utilizado para representar diversos modelos como a distribuição de intensidades de um objeto ou a detecção de transições em vídeos. Na sua forma mais simples, um histograma define-se como uma contagem organizada de informação básica numa sequência de intervalos. Estes intervalos são populados pela contagem de características pré-definidas, como a intensidade dos pixels ou a magnitude do gradiente. Existem alguns métodos automáticos de determinação do *threshold*. O método de Otsu é um dos exemplos mais aplicados [35].

2.2.3 Técnicas de evidenciação de contornos

Nesta secção serão enunciadas algumas transformações de imagem. A ideia fundamental das transformações aqui descritas é a de transformar a imagem numa representação alternativa de informação, de forma a ser mais fácil a extração de algumas características importantes. O chamado enriquecimento de contornos é implementado a partir de filtros espaciais. Segundo [1] existem três principais classes de filtros de melhoramento de contornos:

1. Filtros de deslocamento e diferença (*shift and difference*);
2. Gradiente Prewitt;
3. Filtros Laplacianos.

Para além destes, outros tipos de filtros têm sido desenvolvidos ao longo das últimas décadas.

Sobel: Esta transformação está intimamente ligada ao processo de derivação. A transformação Sobel é aplicada para qualquer ordem de derivação, assim como para combinações de derivadas parciais. Este filtro requer um *kernel* previamente definido. *Kernels* de maiores dimensões apresentam melhores aproximações relativamente ao processo de derivação, pois *kernels* de dimensões pequenas apresentam muita sensibilidade ao ruído.

Canny: Esta transformação de detecção de contornos foi pela primeira vez descrita em 1986, por J. Canny. Nesta transformação, as derivadas de primeira ordem em x e y são calculadas e seguidamente são combinadas em quatro derivadas direcionais. Os pontos constituídos pelos contornos consistem em todos os pontos onde a derivada direcional apresenta máximos locais. A transformação Canny aplica um processo de histerese, definindo os pontos pertencentes aos contornos dos objetos presentes na imagem.

Laplace: A transformação de Laplace é usada frequentemente em diversos contextos. Geralmente, esta transformação é aplicada de forma a facilitar a detecção de objetos. A transformação é representada pela fórmula seguinte:

$$L(f) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Tal como pode ser considerado através desta fórmula, esta transformação está relacionada com um sistema de derivadas de segunda ordem. Considerando um objeto de dimensões reduzidas, rodeado por regiões de elevadas intensidades, é possível verificar que a variação de intensidades imposta faz com que a função, definida pelo operador Laplaciano, seja maximizada. Se o objeto for rodeado por regiões de baixa intensidade, é maximizado o negativo da função considerada. Este facto faz com que esta transformação possa ser utilizada como um eficaz detector de contornos.

2.3 Template matching

Neste capítulo serão introduzidos alguns fundamentos básicos relativos ao método *Template matching*, assim como algumas aplicações e desenvolvimentos realizados nesta área. *Template matching* tem sido um alicerce importante na área do reconhecimento automático de padrões (*pattern recognition*) e ao longo dos anos, vários desenvolvimentos têm sido realizados, sendo hoje uma ferramenta essencial e flexível em diversos sistemas de *computer vision*, entre os quais, sistemas de reconhecimento facial [36]. Um dos principais desafios inseridos neste campo prende-se com o facto de que o reconhecimento facial é uma das tarefas nas quais o Homem apresenta um desempenho praticamente perfeito. Neste sentido, existe a tentativa de que os desenvolvimentos realizados façam com que as máquinas tentem “competir” com a perfeição humana e a realidade é que estes desenvolvimentos têm sido tão significativos que alguns estudos refletem mesmo essa “competitividade” [37].

Template matching consiste em comparar a informação visual extraída de uma imagem com a informação extraída de um *template* [37]. Um *template* é normalmente uma informação visual que contém uma imagem de referência do objeto, ou o conjunto de objetos, a serem detectados na imagem em estudo. É possível definir este método de diversas formas. Este processo é geralmente baseado em aproximações da distância Euclideana ou em modelos de correlação com uma computação espacial de alta velocidade [38][39]. O modelo específico de localização e combinação de objetos depende da complexidade das características da imagem em estudo e do tempo computacional, relacionado com a velocidade de execução. Várias estratégias têm sido estudadas nos últimos anos, tal como introduzido por Marimon et al. em [40].

2.3.1 Transformação de *Hough* (Fundamentos básicos)

Muitos dos modelos desenvolvidos em sistemas de reconhecimento de imagem têm sido baseados na transformada de *Hough*. Nesta secção são introduzidos os fundamentos gerais desta transformação, assim como algumas das aplicações relacionadas.

O objetivo principal desta transformação é a de encontrar determinadas instâncias de objetos específicos, numa determinada imagem, através de determinados tipos de características. A transformação de *Hough* clássica foi criada com o objetivo de identificar segmentos de linha numa imagem. Mais tarde este método foi inovado, de forma a poder identificar formas geométricas arbitrárias (usualmente círculos ou elipses).

Fundamentos teóricos:

O caso mais simples da transformada de *Hough* é a transformação linear de detecção de segmentos de linha. Os segmentos de linha podem ser descritos através da seguinte expressão: $y = mx + b$. Inicialmente é considerada uma transformação em termos dos seguintes parâmetros: um parâmetro relativo ao declive da reta, m e um parâmetro de interseção, b . O segmento de reta pode ser representado como um ponto (b, m) inserido num parâmetro espacial. Tendo em conta o modelo computacional, é preferível utilizar os parâmetros r e θ . O parâmetro r representa a distância entre a linha e a origem e θ representa o ângulo do vector e o ponto mais próximo. Estes parâmetros inserem-se no sistema de coordenadas polares [41].

Através da parametrização, a equação da reta é seguidamente descrita:

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r}{\sin\theta}\right)$$

A equação pode sofrer o seguinte rearranjo:

$$r = x\cos\theta + y\sin\theta$$

O plano aqui descrito pode ser referido como o espaço de *Hough* para o conjunto de segmentos de linha. Para cada linha é possível associar o par (r, θ) , que é único desde que:

$$\theta \in [0, \pi]; r \in \mathbb{R}$$

Considerando um conjunto de pontos, para encontrar um conjunto de segmentos de linha que se ajustem a estes mesmos pontos, é executada a transformação dos pontos (x_i, y_i) nas curvas sinusoidais no plano (r, θ) , definida por:

$$r = x_i\cos\theta + y_i\sin\theta$$

As curvas correspondentes a pontos colineares têm um ponto comum de interseção. Este ponto no plano (r, θ) define a linha que passa sobre o ponto colinear e assim, o problema de detecção de pontos colineares pode ser convertido num problema de detecção de curvas concorrentes. Duda e Hart descrevem em [41] algumas propriedades desta transformação, tais como:

1. Um ponto no plano da imagem corresponde a uma curva sinusoidal no plano parametrizado.
2. Um ponto no plano parametrizado corresponde a uma reta no plano da imagem.
3. Pontos sobrepostos na mesma reta no plano da imagem correspondem a curvas, através de um ponto comum, no plano parametrizado.
4. Pontos sobrepostos na mesma curva no plano parametrizado correspondem às linhas, através de um ponto comum, no plano da imagem.

A maioria dos algoritmos que têm sido implementados utilizam, geralmente, um sistema de acumulação, de forma a detectar a existência de segmentos de reta. Este sistema representa-se por um vetor com uma dimensão igual ao número de parâmetros desconhecidos. As duas dimensões do vetor deverão corresponder aos valores quantizados (r, θ) . O algoritmo implementado determina se existem evidências suficientes de um contorno para cada pixel da imagem em consideração. Se sim, irá calcular os parâmetros do segmento correspondente e incrementar o acumulador no intervalo correspondente ao parâmetro evidenciado. Seguidamente o máximo local do acumulador é determinado e os segmentos mais semelhantes serão extraídos.

As bases teóricas descritas acima são aplicadas apenas a segmentos de reta. Num processo de detecção de outras formas geométricas, que podem ser representadas com um conjunto de parâmetros específicos, uma transformação similar pode ser utilizada. Círculos, elipses ou curvas arbitrárias, por exemplo, podem ser transformadas em conjuntos de três parâmetros, representando o centro e o raio. Para formas simples, a transformada de *Hough* pode ser representada por uma função modelo definida da seguinte forma [42]:

$$f(x_1, x_2, \dots, x_d; p_1, p_2, \dots, p_m) = 0$$

$X = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d$ assume-se como um ponto contido no conjunto de entrada e $P = (p_1, p_2, \dots, p_m) \in \mathbb{P}^m$ é um vetor contido no espaço parametrizado \mathbb{P}^m . Cada parâmetro do vetor P em \mathbb{P}^m caracteriza a forma do objeto. A transformada de *Hough* determina cada ponto x através de um mapeamento, determinando todas as instâncias do objeto sobre o ponto x . O processo de mapeamento consiste em definir um subconjunto pré-definido de k parâmetros em P (para $k < m$) e determinar os restantes $(m - k)$ parâmetros, utilizando uma função de mapeamento na forma: $\{p_{k+1}, p_{k+2}, \dots, p_m\} = g(x_1, \dots, x_d; p_1, \dots, p_k)$. Este processo é obtido através da função modelo, incluída na equação expressa acima. O vetor P associado é seguidamente anexado a uma determinada posição de um vetor acumulador definido. O valor guardado na posição é incrementado e no fim do processo, as posições com um maior número de valores correspondem aos parâmetros que indicam a respetiva correspondência com as formas geométricas em análise.

Fernandes e Oliveira apresentam em [42] um método baseado na transformação de *Hough* que é aplicado na detecção de formas analíticas. Neste trabalho é descrita uma aproximação geral de detecção de segmentos num sistema de informação multidimensional com grande quantidade de ruído. A aproximação é independente das propriedades geométricas dos segmentos a serem detectados, assim como independente do tipo de informação de entrada, sendo definida uma solução que parametriza qualquer subespaço. O procedimento de mapeamento utilizado é independente do tipo de informação de entrada e o método implementado apresenta resultados satisfatórios em vários sistemas de informação real.

Modelos alternativos:

A transformada de *Hough* pode ser processada depois da aplicação de vários filtros de pré-processamento, de forma a aumentar a qualidade dos contornos. O’Gorman e Clowes propuseram um modelo que detecta segmentos de reta, tendo em conta o gradiente local da intensidade da imagem [43]. Considerando que um dado ponto (x,y) é intersetado por uma reta, a direção local do gradiente determina os parâmetros θ e r correspondentes à reta. Este modelo permite a redução do tempo de computação e do número de votos úteis.

Uma transformação inovadora baseada na optimização de um vetor acumulador foi introduzida por Ji et al. em [44]. Os desenvolvimentos realizados permitem detectar segmentos de reta de forma bastante eficiente em imagens com regiões de fundo mais complexas. Um operador local é implementado de forma a melhorar a diferença entre os pontos de acumulação e o ruído presente na imagem. Seguidamente é obtido um *threshold* global de forma a que sejam analisados os picos anteriormente detectados.

Xie e Ji apresentaram um novo método de detecção de elipses [45]. Este método apenas necessita de um vetor acumulador de uma dimensão. Este vetor acumula as informações relativas ao comprimento do eixo menor da elipse. O algoritmo implementado não necessita da avaliação de tangentes ou curvaturas dos contornos das elipses, que são geralmente muito sensíveis às condições de ruído. O algoritmo apresenta uma implementação simples, não necessitando de uma computação matemática demasiadamente pesada.

Têm sido desenvolvidas várias aplicações da transformada de *Hough* em processamento de imagem microscópico. Um dos vários estudos foi desenvolvido por Bewes et al. em [29]. Eles apresentaram um método automático de contagem de colónias e utilizam uma transformação completa de *Hough* no seu algoritmo. O algoritmo proposto apresenta três passos principais: 1-pré-processamento; 2-processamento utilizando uma transformação de *Hough* circular e generalizada e 3-pós-processamento. Estes passos de processamento são incorporados num programa denominado de *CHiTA* (*circular Hough image transform algorithm*).

A transformada de *Hough* é aplicada através de um processo de votação, relacionado com uma equação circular. O algoritmo apresenta um elevado grau de eficiência contra artefatos, ruído generalizado e não necessita de hardware especializado.

2.3.2 *Template matching* – algoritmos de correlação

Os problemas de correspondência dependem de vários fatores de transformação, como translação, rotação, escalamento ou inclinação da imagem, de forma a proceder a uma posterior definição dos *templates* a serem processados [46]. Os *templates* são usados num processo de correspondência com as imagens em consideração, sendo o processo de correspondência baseado em algoritmos matemáticos que avaliam a similaridade entre determinadas regiões da imagem e o *template*. Esta similaridade pode ser calculada de diversas formas. Seguidamente serão introduzidos alguns dos principais algoritmos de correlação onde R representa a imagem resultante do processo; (x, y) define as coordenadas; w e h definem, respetivamente, a largura e a altura de cada elemento, I representa a imagem e T representa o *template*.

Método de correspondência: diferença quadrática

Este método processa a diferença quadrática entre a imagem e o *template*. Uma correspondência perfeita corresponderá ao valor 0 e uma má correspondência corresponderá a um valor muito elevado.

$$R_{sqdiff}(x, y) = \sum_{x', y'} [T(x', y') - I(x + x', y + y')]^2$$

Método de correspondência: método de correlação

Este método processa a correspondência entre a imagem e o *template*, aplicando um processo multiplicativo. Uma correspondência perfeita corresponderá a um valor elevado e uma má correspondência corresponderá ao valor 0.

$$R_{ccorr}(x, y) = \sum_{x', y'} [T(x', y') \cdot I(x + x', y + y')]$$

Método de correspondência: coeficiente de correlação

Este método processa a correspondência do *template* com uma imagem, relativamente ao valor médio dos seus valores, respetivamente. Uma correspondência perfeita corresponderá a ao valor 1 e uma correspondência inversa corresponderá ao valor -1. O valor 0 significa que não existe correlação.

$$R_{ccoeff}(x, y) = \sum_{x', y'} [T'(x', y') \cdot I'(x + x', y + y')]$$

$$T'(x', y') = T(x', y') - (w \cdot h) \frac{1}{\sum_{x'', y''} T(x'', y'')}$$

$$I'(x + x', y + y') = I(x + x', y + y') - (w \cdot h) \frac{1}{\sum_{x'', y''} I(x + x'', y + y'')}$$

Métodos normalizados:

Existem também técnicas de normalização para cada um dos métodos descritos acima. O processo de normalização reduz algumas diferenças de iluminação que possam existir entre o *template* e a imagem.

$$R(x, y) = \sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}$$

Os métodos aqui descritos são devidamente explicitados em [47], assim como algumas das suas propriedades. Alguns processos alternativos são descritos em [48]. Aqui são aplicadas algumas técnicas baseadas nas características implícitas da imagem em consideração. Um número fixo de características é extraído e representado na forma de um vetor. A correspondência entre o *template* e a imagem é feita através de um processo de comparação entre os vetores criados. Esta estratégia foi adotada em vários estudos e apresenta resultados satisfatórios, quando comparados com alguns algoritmos de correspondência "pixel-por-pixel".

2.3.3 Template matching – aplicações gerais

As técnicas de *Template matching* desenvolvidas assumem-se como uma parte importante de muitos sistemas *computer vision*. O processo de desenvolvimento de um sistema para uma tarefa específica requer muitas vezes a projeção de aproximações de correspondência alternativas e ao longo dos anos vários estudos têm sido feitos neste sentido.

Choi et al. propuseram um algoritmo, invariante em termos de rotação, baseado na combinação de métodos de projeção e momentos de Zernike [38]. O método consiste, numa primeira fase, numa seleção dos pontos de maior correspondência, através de alguns parâmetros característicos e seguidamente, na aplicação de um processo baseado em momentos de Zernike. Lowe et al. apresentaram um método de extração de características distintivas, em imagens utilizadas para executar um processo de correspondência, contendo diferentes perspectivas do mesmo objeto [49]. Estas características apresentam várias propriedades que auxiliam a eficiência de todo o processo, sendo este invariante em termos de escalamento, rotação e parcialmente invariante em termos de iluminação. Este modelo apresenta potenciais aplicações, incluindo sistemas de reconstrução 3D, localização robótica, sistemas de seguimento de movimentos e em muitas outras aplicações.

Berg et al. apresentaram uma técnica de *Template matching* com o objetivo de encontrar pontos de correspondência na presença de regiões de elevada distorção [50]. Este método é baseado na definição de características relacionadas com os contornos que definem os objetos presentes nas imagens em consideração. Fredriksson et al. propuseram uma forma de localizar um conjunto de pontos com alta probabilidade de correspondência [40]. Este método é baseado num descritor de orientação invariante. Ullah et al. introduziram um código de correspondência baseado na aplicação de histogramas [51]. Os histogramas são utilizados de forma a estimar a orientação de segmentos em cada ponto da imagem. Marion et al. apresentaram um método de *Template matching* invariante em rotação [40]. O modelo descrito

é introduzido em três passos. Inicialmente a magnitude do gradiente é comparada de forma a que seja processada uma rápida localização de pontos com alta probabilidade de correspondência. Seguidamente, o resultado do primeiro passo é melhorado pela aplicação de histogramas de gradiente. Por último, o processo de *Template matching* é aplicado apenas aos pontos de elevada magnitude de gradiente e aos pontos de maior similaridade. Esta similaridade é avaliada através da análise dos histogramas obtidos. Este método é processado através da determinação da correlação normalizada entre a intensidade dos pontos vizinhos presentes na imagem e o *template*.

Omachi et al. apresentou um algoritmo inovador que utiliza um polinómio de Legendre que apresenta uma aproximação ao *template*, de forma a que seja executada a correspondência entre o polinómio e a imagem, em vez de ser feita a correspondência entre o *template* e a imagem [39]. Este algoritmo apresenta um elevado grau de eficiência, especialmente quando as dimensões do *template* diferem das dimensões da imagem a ser processada. Esta eficiência traduz-se também em termos de custo computacional. Outros desenvolvimentos têm sido feitos de forma a aproximar a imagem a uma determinada função, reduzindo o custo computacional. Schweitzer et al. introduziram um algoritmo *Template matching* que utiliza uma aproximação à imagem através de um polinómio de segunda ou terceira ordem [52].

O modelo SIFT (*Scale Invariant Feature Transform*) é um dos mais robustos algoritmos de características invariantes. Esta técnica foi inicialmente desenvolvida para imagem em escala de cinzentos. Atualmente existem algumas aproximações na literatura que utilizam a cor como principal característica [53]. Lowe desenvolveu um sistema de reconhecimento de objetos que utiliza uma nova classe de características locais [54][49]. O primeiro passo deste algoritmo identifica localizações chave no espaço escalar, através da procura de pontos extremos, obtidos por intermédio da aplicação de uma função Gaussiana. Seguidamente são gerados vetores nos quais são incluídas determinadas características das imagens em consideração. Estes vetores são denominados de *SIFT Keys*. O processo de reconhecimento utiliza um algoritmo *nearest-neighbor* de rápida execução, seguida por uma transformação de *Hough* que identifica grupos pertencentes a objetos singulares.

Um dos mais eficientes algoritmos de *Template matching* foi introduzido por Bay et al. em [55], assumindo-se como um inovador algoritmo de detecção, invariante em termos de rotação e escalamento. Este método utiliza uma aproximação baseada numa matriz de Hessian, estabelecendo um bom compromisso entre a complexidade e a robustez das características em causa. Para além da detecção de objetos, SURF (*Speed-Up Robust Features*) tem sido aplicado a imagens da retina humana, sendo inserido um modelo de reconstrução 3D. Este método apresenta uma elevada velocidade computacional, em comparação com outros métodos, devido à utilização de imagens integrais, que reduzem o número de operações efetuadas.

Alves de Araújo e Kim desenvolveram em [56][57] um método baseado num modelo *Template matching* invariante em termos de rotação, escalamento, translação, iluminação e contraste, denominado de *Ciratefi* (*Circular, Radial and Template-Matching Filter*). Os desenvolvimentos posteriormente realizados permitem a extensão deste modelo a imagens com cor. O objetivo do *Ciratefi* é o de encontrar ocorrências de um *template* numa imagem a ser analisada, com uma dada orientação angular e escalar. Este algoritmo consiste em três filtros sucessivos: *Cifi*, *Rafi* e *Tefi*, que sucessivamente excluem determinados pixels que não possuem as características necessárias para serem incluídos no processo de correspondência. Uma das grandes vantagens deste algoritmo é que não necessita de uma simplificação prévia,

quer por parte do *template*, quer por parte da imagem a ser analisada. Neste sentido, *Ciratefi* não necessita de informação adicional, relativa à escala de cinzentos, referente à detecção de contornos, detecção de pontos de interesse, ou processo de binarização. Isto faz com que este modelo apresente um algoritmo muito robusto. É implementado um método de correlação normalizada em cada passo, de forma a avaliar a eficiência da correspondência do *template* com cada região da imagem.

O modelo *Ciratefi* apresenta resultados mais satisfatórios do que o modelo *SIFT* nos casos em que as imagens apresentam uma desfocagem maior ou apresentam grandes variações em termos de iluminação e contraste. *SIFT* é mais eficiente do que o *Ciratefi* em imagens que apresentam uma maior variação em termos de textura e quando seguido de uma transformada de *Hough*, mostra-se mais robusto relativamente a oclusões parciais. *SIFT* é mais rápido do que o *Ciratefi* na determinação das correspondências para uma dada imagem, pois a maior parte do tempo de processamento é gasta no processo de pontos chave e de extração de características da imagem.

2.3.4 *Template matching* – aplicações em microscopia

Várias técnicas de *Template matching* têm sido aplicadas a imagens de microscopia na última década. Existem diversos algoritmos que permitem a execução da identificação de formas circulares ou elípticas e que apresentam resultados satisfatórios no reconhecimento de formas geométricas no tipo de imagem considerado. O processo de quantificação de bactérias em imagens microscópicas é visto como um dos exemplos dos vários estudos realizados em microrganismos. A optimização deste tipo de procedimento proporciona um desenvolvimento de várias aproximações de identificação automática das partículas existentes numa determinada amostra. Roa et al. apresentaram um algoritmos que processa uma caracterização automática de bactérias [58]. A aproximação proposta consiste em quatro passos progressivos: 1-melhoramento da imagem em análise; 2-aplicação de um algoritmo de segmentação baseado no método *Watershed*; 3-optimização do processo de segmentação inicial e 4-quantificação das bactérias através de um procedimento de *Template matching*. Os *templates* utilizados são construídos através de padrões definidos de bactérias, que consideram um conjunto de formas e orientações, com o intuito de incluir uma elevada percentagem de variações, em termos das características das bactérias presentes. Um método de segmentação de imagem através de um processo de reconhecimento de determinadas formas e baseado nas orientações dos contornos foi apresentado por Teutsch et al. em [59]. Este modelo é aplicado em diversos tipos de imagens, nomeadamente em imagens contendo colónias de bactérias. O processo de detecção apresenta resultados bastante satisfatórios. Souza et al. propuseram um método de correlação normalizada que executa uma contagem automática de colónias de bactérias [60]. Bewes et al. também apresentaram um algoritmos de contagem automática de colónias, baseado em reconhecimento de imagem (já descrito na **secção 2.3.1**)[29].

2.4 Watershed

Tal como foi já anteriormente apresentado, os fundamentos do método *Watershed* têm sido desenvolvidos durante décadas, sendo uma das técnicas mais aplicadas em modelos de segmentação de imagens microscópicas. Nesta secção será feita uma abordagem relativa aos primeiros estudos realizados e à forma como esta técnica de segmentação têm sido inovada ao longo dos anos. Além disto, serão abordados alguns dos fundamentos básicos relativos a esta técnica de segmentação.

Num breve olhar em direção aos primórdios, é possível observar que os conceitos de *Watershed* estão muito relacionados com o campo da topologia. Em [61] o processo básico do método *Watershed* é explicado de uma forma bastante intuitiva. Os autores deste artigo dão o exemplo de uma linha particular que separa os Estados Unidos em duas regiões, chamada de “Grande divisão” (“*Great divide*”). Sendo considerada uma queda de água sobre um dos lados da linha, o fluxo de água pode descer até alcançar o oceano Atlântico. Se a queda de água acontecer do outro lado da linha, o fluxo pode descer até alcançar o oceano Pacífico. Deste modo, esta linha constitui um típico exemplo de uma linha *Watershed*, que pode ser vista como uma barragem e as duas regiões separadas como bacias hidrográficas. Os dois oceanos são considerados os pontos mínimos associados a cada bacia. Pode-se facilmente transferir este raciocínio para o campo do processamento de imagem. Imagens em escala de cinzentos podem ser consideradas como superfícies topológicas. Em termos de imagem digital, as bacias consistem em largos grupos de pixels conectados. Assim existirão pontos designados de máximos locais, em que a queda de água terá igual probabilidade de cair em duas bacias adjacentes.

O método *Watershed* é geralmente utilizado para separar objetos unidos entre si, uma das operações mais complexas em processamento de imagem [4][62]. Este método assume-se como sendo uma ferramenta para a segmentação de imagem, apresentando vantagens relativamente a outras técnicas de segmentação. As vantagens da *Watershed* sobre métodos baseados em detecção de contornos, é que geralmente reproduz contornos próximos que definem eficazmente os limites dos objetos em análise. Esta aproximação insere-se numa classe denominada de *region based techniques*, na qual são considerados processos de similaridade entre determinadas regiões e os pixels constituintes [63].

A transformação *Watershed* foi inicialmente introduzida como uma ferramenta morfológica por Digabel e Lantuejoul em [64]. Neste estudo foram analisadas várias imagens binárias, resultantes de sucessivos processos de *threshold* aplicados a várias imagens. Uns anos mais tarde, Beucher e Lantuejoul propuseram um algoritmo de *Watershed* através de um processo de imersão. Este modelo é baseado na determinação de zonas de influência que correspondem à definição dos diferentes níveis de intensidade da superfície em estudo. Este algoritmo poderá apresentar resultados inesperados, como arcos indesejados, devido a questões relativas à homotopia da superfície resultante [61]. Beucher também apresentou vários estudos que relacionam o modelo *Watershed* com processos de esqueletização, sendo que as linhas *Watershed* podem ser determinadas através da esqueletização dos tons de cinzento.

Seguidamente serão introduzidos alguns fundamentos básicos relativos à técnica *Watershed*, assim como a algumas ferramentas utilizadas.

2.4.1 Watershed – Fundamentos básicos

Gradiente morfológico

Um dos principais critérios de segmentação prende-se com a homogeneidade das intensidades contidas em cada objeto presente nas imagens. Neste sentido, o gradiente da imagem é frequentemente utilizado em transformações *Watershed* [63].

O gradiente morfológico de uma imagem pode ser definido da seguinte forma.

$$g(f) = (f \oplus B) - (f \ominus B)$$

Onde $(f \oplus B)$ e $(f \ominus B)$ são, respetivamente, as operações elementares de dilatação e erosão de f (f é a função que representa uma determinada imagem em escala de cinzentos; B representa uma mascara específica e $g(f)$ representa a função de gradiente). O gradiente da imagem é obtido através da diferença algébrica entre uma operação de dilatação e uma operação de erosão [65].

Função distância

Nos casos em que a forma geométrica de um objeto é um dos principais critérios no processo de segmentação, a função distância pode ser bastante útil.

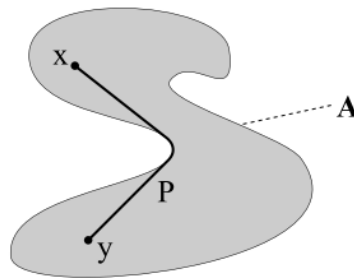


Figura 2.1: Distância entre x e y no interior do conjunto A (retirado de [59]).

Considerando Y um subconjunto de X é possível determinar o conjunto de todos os pontos de X que se encontram a uma distância finita de Y :

$$R_X(Y) = \{x \in X : \exists y \in Y, d_X(x, y) \text{ finito}\}$$

$R_X(Y)$ é denominado pelo conjunto reconstruído em X através do subconjunto Y . É composto por todos os pontos conectados de X que são marcados por Y .

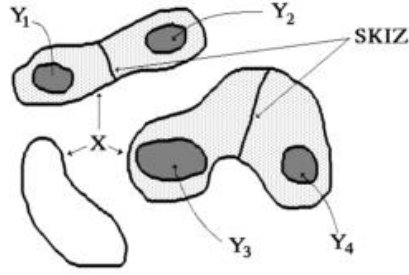


Figura 2.2: Processo de esqueletização num dado conjunto X (retirado de [57]).

Esta definição está na base de muitos operadores morfológicos e foi introduzida no processo de análise imagiológica por Beucher e Lantuejoul em [66].

Considerando $X \subset \mathbb{Z}^2$ um conjunto de \mathbb{Z}^2 e $Y \subseteq X$. O processo de dilatação geométrica de dimensões $n \geq 0$ de Y em X é o conjunto de pixels de X cuja distância a Y é menor ou igual a:

$$\delta_X^{(n)}(Y) = \{p \in X \mid d_X(p, Y) \leq n\}$$

Se Y for composto por n componentes Y_i , a zona de influência $z_X(Y_i)$ de Y_i é o conjunto de pontos de X a uma distância finita de Y_i e mais próximos de Y_i do que de qualquer outro Y_j :

$$z_X(Y_i) = \{x \in X : d_X(x, Y_i) \text{ finito and } \forall j \neq i, d_X(x, Y_i) < d_X(x, Y_j)\}$$

Os limites entre as várias zonas de influência determinam o processo de esqueletização por zonas de influência (SKIZ) de Y em X .

$$SKIZ_X(Y_i) = X / z_X(Y_i)$$

$$z_X(Y) = \bigcup z_X(Y_i)$$

Mínimos e máximos de uma função

Os mínimos e máximos locais de uma dada função apresentam propriedades que poderão ser úteis em vários sistemas computacionais. O conjunto de todos os pontos $\{x, f(x)\}$ pertencentes a $\mathbb{Z}^2 \times \mathbb{Z}$ podem ser vistos como uma superfície topográfica S .

Considerando dois pontos s_1 e s_2 da superfície S . Um caminho entre $s_1(x_1, f(x_1))$ e $s_2(x_2, f(x_2))$ pode ser apresentado como qualquer sequência $\{s_i\}$ de pontos contidos em S , com s_i adjacente a s_{i+1} . Um caminho não ascendente apresenta-se como:

$$\forall s_i(x_i, f(x_i)), s_j(x_j, f(x_j)) \quad i \geq j \leftrightarrow f(x_i) \leq f(x_j)$$

Um ponto $s \in S$ pertence a um mínimo se não existirem caminhos ascendentes iniciados em s . Um mínimo pode ser considerado como uma bacia contida na superfície topológica. O conjunto M de todos os mínimos de f é constituído por vários componentes ligados $m_i(f)$. Uma definição similar é definida para os pontos máximos.

Algoritmo *Watershed*: Fundamentos básicos

Seguidamente será descrito um dos diversos algoritmos de *Watershed* já desenvolvidos. Considere-se um algoritmo baseado num processo de *Watershed* por imersão. Inicialmente é definida uma função f que representa uma determinada superfície topográfica. Cada região correspondente aos vários mínimos locais $m_i(f)$ é imersa com uma velocidade constante. Durante esta imersão, o nível sobe em cada uma dessas regiões, sendo que entre duas ou mais bacias, o nível pode convergir. De forma a evitar esta situação, são construídas barreiras nos pontos de convergência (ver Figura 2.3). Estas barreiras definem o método *Watershed* resultante a partir da função f e separam cada bacia, contento um e apenas um mínimo local [61].

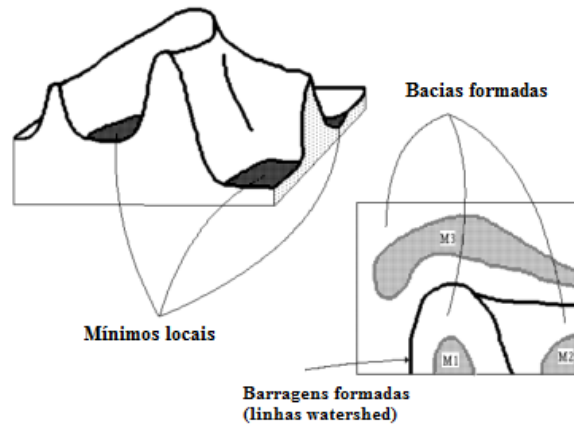


Figura 2.3: Processo de formação das linhas *Watershed* (adaptado de [57]).

Formalmente este processo pode ser expresso da seguinte forma. Considerando f a imagem em estudo, i_{min} e i_{max} respetivamente, o menor e o maior valor de f , num domínio D_f . Considerando $T_i(f)$ o *threshold* de f a um nível i :

$$T_i(f) = \{p \in D_f, f(p) \leq i\}$$

Cada bacia associada a um mínimo local m é representada por $C(m)$ e o subconjunto desta bacia constituído pelos pontos que apresentam uma altitude menor ou igual a i , $C_i(m)$.

$$C_i(m) = \{p \in C(m), f(p) \leq i\} = C(m) \cap T_i(f)$$

A simulação da imersão começa com a definição do conjunto de pontos $T_{i_{min}}(f)$, que correspondem aos primeiros pontos a serem “preenchidos”. Considerando $X_{i_{min}}$ os pontos de f que pertencem aos mínimos de menor altitude, define-se que:

$$X_{i_{min}} = T_{i_{min}}(f)$$

Sendo Y um dos componentes ligados de $T_{i_{min}+1}(f)$, onde $X_{i_{min}} \subseteq T_{i_{min}+1}(f)$, existem três possíveis relações entre Y e $Y \cap X_{i_{min}}$:

- A. $Y \cap X_{i_{min}} = \emptyset$: neste caso, Y é obviamente um novo mínimo de f . Todos os pixels vizinhos não pertencem a $T_{i_{min}+1}(f)$ e apresentam uma intensidade ligeiramente maior do que i_{min+1} .

- B. $Y \cap X_{i_{min}} \neq \emptyset$ e apresenta ligação: neste caso, Y corresponde exatamente aos pixels pertencentes à bacia associada ao mínimo $Y \cap X_{i_{min}}$ e apresenta uma intensidade menor ou igual a i_{min+1} .
- C. $Y \cap X_{i_{min}} \neq \emptyset$ e não apresenta ligação: Y contém diferentes mínimos de f . Considerando Z_1, Z_2, \dots, Z_k estes mesmos mínimos, neste ponto, a melhor escolha para $C_{i+1}(Z_a)$ é dada pela zona de influência de Z_a contida em Y :

$$C_{i_{min}+1}(Z_a) = az_Y(Z_a)$$

Esta relação aplica-se a todos os níveis de i . O segundo conjunto desta recursão é dado por:

$$X_{i_{min}+1} = \min_{i_{min}+1} \cup fZ_{T_{i_{min}+1}}(X_{i_{min}})$$

O conjunto de bacias da imagem f é igual ao conjunto $X_{i_{max}}$, obtido depois da seguinte recursão:

1. $X_{i_{min}} = T_{i_{min}}(f)$;
2. $\forall i \in [i_{min}, i_{max} - 1], X_{i+1} = \min_{i+1} \cup fZ_{T_{i+1}(f)} \cdot (X_i)$.

O resultado do método *Watershed* sobre a imagem f corresponde ao conjunto de pontos de D_f que não pertence a nenhuma das bacias. O processo de recursão relativo a dois níveis sucessivos é seguidamente ilustrado.

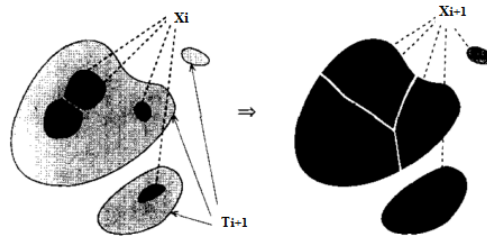


Figura 2.4: Processo de recursão entre X_i e X_{i+1} (adaptado de [55]).

O algoritmo descrito acima é um exemplo que simula o processo de um algoritmo *Watershed*. Existem, no entanto, vários algoritmos alternativos a este. Beucher descreveu um algoritmo baseado na determinação de setas orientadas que representam uma determinada função [63]. Considerando $f: \mathbb{Z}^2 \rightarrow \mathbb{Z}$ essa mesma função, é definido um grafo orientado, cujos vértices são pontos \mathbb{Z}^2 , que apresentam setas desde x até um ponto adjacente y e que satisfaz a condição $f(x) < f(y)$.

2.4.2 Aplicações gerais

Gradiente da imagem

Uma das aproximações possíveis no processo de segmentação de imagem é a extração do conjunto de contornos de cada objeto presente na imagem em análise. O processo de *Watershed* sobre o gradiente da imagem esta na base de diversos algoritmos de segmentação. Um dos critérios de segmentação prende-se com a homogeneidade da intensidade dos pixels presentes em cada objeto. Neste sentido, um simples uso de uma técnica de *threshold* pode não ser suficiente na aplicação de um processo de *Watershed*, sendo que a aplicação de uma função gradiente corresponderá a resultados mais satisfatórios [63]. Uma aplicação direta deste processo pode provocar situações de sobressegmentação, constituindo na maioria das vezes uma computação demasiadamente pesada. Este facto pode ocorrer devido a um nível elevado de ruído ou devido às características inconstantes das imagens em estudo [65]. Muitos algoritmos têm sido desenvolvidos de forma a evitar este facto. Soille, por exemplo, apresentou uma ferramenta inovadora, baseada na determinação de um gradiente morfológico que é determinado através do cálculo de gradientes direcionais [61]. Vachier et al. apresentaram um estudo progressivo na aplicação de técnicas *Watershed* em imagens de gradiente. De forma a evitar os problemas de sobressegmentação, é desenvolvido um sistema alternativo no processo de imersão inserido no algoritmo desenvolvido. Este processo é combinado com uma determinação de marcas que irão iniciar todo o processo de segmentação [62].

Sistema de marcas

Na aplicação de um algoritmo de *Watershed*, muitos estudos mostram que nos resultados finais, alguns segmentos resultantes podem corresponder a objetos que não deveriam ser extraídos, nos quais os respetivos contornos não deveriam ser visualizados na imagem segmentada. Neste caso, uma das possibilidades é a de modificar a função de forma a que as bacias resultantes apenas correspondam aos objetos desejados. Estes métodos necessitam de um passo inicial de determinação de marcas [67]. Uma marca é definida como um conjunto conectado de pixels incluídos numa região singular e específica. O conjunto de marcas pode ser determinado manualmente ou automaticamente. A grande parte dos sistemas desenvolvidos depende de um método de determinação automática de marcas, devido, principalmente, à importância de um modelo que não seja excessivamente pesado em termos de tempo de execução e em termos de recursos utilizados [68][69]. A ideia fundamental é a de criar um método que detecta uma marca para cada segmento da imagem considerada [70]. À região do fundo também é definida uma marca que irá corresponder à região de menor intensidade da imagem em estudo. Depois das marcas serem extraídas, estas são definidas como pontos mínimos de uma função gradiente e os mínimos locais que não correspondem às marcas, não são tidos em conta, através de um processo de imersão das bacias correspondentes [62].

Vários algoritmos têm sido desenvolvidos de forma a estabelecer um processo de determinação automática e eficiente de marcas. Grimaud et al. descreveram um processo dinâmico baseado em operadores específicos. Neste trabalho é proposta uma transformação que avalia a determinação de pontos mínimos, através de critérios específicos de contraste [71].

O método *Watershed* assume-se como uma ferramenta mais robusta e eficiente, quando inclui uma definição correta de marcas, o que é provado por vários estudos já

realizados. Este processo é facilmente aplicado a várias aplicações e a diversos tipos de processos de segmentação [62][61].

Problemas de sobressegmentação

A sobressegmentação é uma das principais barreiras na aplicação do método básico de *Watershed*. A aplicação desta técnica apresenta geralmente demasiadas bacias sendo associado a cada uma um mínimo local. A sobressegmentação é uma consequência do excesso de pontos mínimos, que são geralmente produzidos por variações impostas pelo ruído presente nas imagens em estudo. O processo de determinação de marcas, descrito acima, ajuda a prevenir essa situação. Para além disto, alguns algoritmos alternativos com processos variacionais de definição dos mínimos locais foram já desenvolvidos [63][62].

2.4.3 Aplicações em imagens microscópicas

A transformação *Watershed* é uma técnica que ao longo dos anos tem sido muito aplicada no processamento de imagens de microscopia. Ao longo dos anos vários algoritmos têm sido introduzidos com o objetivo de identificar e caracterizar alguns tipos de organismos. Alguns dos estudos realizados são seguidamente descritos.

Dow et al. desenvolveram um algoritmo baseado num método de *Watershed* para segmentar imagens de núcleos *Hoechst-stained* com uma exatidão de 85%. Este algoritmo determina uma única marca para cada núcleo presente na imagem, define uma imagem que compensa as situações de elevado ruído e sobressegmentação e finalmente, localiza as linhas divisórias de grande intensidade [72].

Adiga propôs uma extensão a uma técnica *Watershed* aplicada a três dimensões que em vez de encontrar uma marca para cada objeto numa fase inicial, permite a identificação dos fragmentos, através de critérios como a dimensão, a forma e a conectividade espacial. O primeiro passo é complementado com uma identificação dos vários sobre-segmentos resultantes [73]. Alguns anos mais tarde, Adiga et al. apresentaram um software de análise integrada de imagens, de forma a processar a segmentação de núcleos. Foi criada uma técnica *region-growing* com uma definição automática de marcas (sementes). As características dos núcleos são fatores que influenciam a execução desta técnica [74]. Lin et al. descreveram um algoritmo 3D que combina uma técnica de *Watershed* clássica com outras técnicas que exploram os fundamentos de gradiente de intensidade e várias questões anatómicas dos núcleos considerados [75]. A sobressegmentação é evitada através de um processo de pós-processamento. Este processo localiza os contornos dos segmentos e utiliza algumas técnicas de junção de segmentos, baseada em modelos de informação dos objetos resultantes.

Um método de segmentação aplicado a glóbulos brancos é proposto por Jiang et al. em [76]. Esta técnica utiliza um filtro de escalamento espacial e uma técnica de *Watershed* a três dimensões que não é influenciado por determinadas características das imagens em estudo e extrai as regiões constituídas por conjuntos de glóbulos brancos de uma forma bastante efetiva.

Selinummi et al. desenvolveram um software de quantificação de bactérias em imagens digitais. Depois de uma correcção da região de fundo, devido a variações de iluminação, os pixels são separados da região de fundo através de um processo de *threshold* global. As

bactérias inseridas nas várias colónias são separadas através de um processo de *Watershed* baseado na intensidade das células consideradas [77].

Uma aproximação diferente foi estudada por Chen et al. num processo de quantificação do ciclo celular em células individuais [78]. Foi implementado um algoritmo baseado no método de Otsu, sendo aplicado um processo de binarização. Seguidamente foi implementado um algoritmo de *Watershed* que corrige algumas situações de sobressegmentação. Wang et al. descreveram uma aproximação com alguns pontos similares, de forma a processar uma identificação automática de células capturadas através de processos de microscopia por fluorescência [79]. Neste estudo é proposta uma detecção celular baseada na intensidade e na forma dos objetos constituintes. Uma extracção de informação é considerada através de um processo de binarização. Seguidamente a informação relativa tanto à intensidade como à forma, é adquirida num processo de geração de máximos locais. Por fim os pontos centrais das células são detectados e utilizados num posterior processo de segmentação através de uma técnica de *Watershed*. Zhou et al. propuseram um método de *threshold*, seguido por uma técnica de *Watershed*, de forma a processar a segmentação, o seguimento e a quantificação de ciclos celulares de uma larga população de células [80]. O método *Watershed* é aplicado de forma a separar as várias colónias de células e seguidamente é aplicado um processo de junção de fragmentos que combina critérios de distribuições de probabilidades e de compactação celular que reduzem as situações de sobressegmentação.

Um método inovador foi desenvolvido por Ates com o objetivo de criar um sistema de contagem automática de colónias de bactérias [28]. Inicialmente é implementada uma binarização simples que utiliza um sistema de determinação automática de *thresholds*, baseado em histogramas. Os objetos são analisados de acordo com o tamanho e o raio de compactação, de forma a determinar a similaridade entre as diversas formas encontradas e as formas circulares, consideradas ideais. Um algoritmo *Watershed* é seguidamente aplicado com o objetivo de dividir as formas correspondentes, nos locais onde o teste de compactação falha. Havendo a hipótese de o resultado não corresponder na sua maioria a colónias circulares, o teste volta a ser executado. Se os resultados voltarem a não ser satisfatórios, é aplicada uma ferramenta alternativa. Ao et al. desenvolveram um algoritmo semiautomático de segmentação, de forma a proceder a uma avaliação da estrutura nuclear de um conjunto de células [81]. Inicialmente é gerado um “perfil celular” através de um algoritmo *Watershed* baseado em técnicas *k-means*. O “perfil celular” define a separação do citoplasma e núcleo celulares da região de fundo. É depois aplicado um processo de *threshold*, baseado em histogramas, de forma a isolar os núcleos. Por fim, é aplicada uma técnica *Watershed* baseada em técnicas de *clustering*.

Vários modelos *Watershed* baseados nas propriedades do gradiente são aplicados em imagens microscópicas. Lin et al. propuseram uma aproximação baseada numa técnica de *Watershed*, controlada pelo nível do gradiente da imagem que inclui um sistema de marcação automática [82]. Este modelo foi utilizado para isolar automaticamente células individuais a partir de uma biopsia virtual em THG (*Third Harmonic Generation*). Roa et al. aplicaram um algoritmo *Watershed* por imersão, baseado na aplicação de técnicas de gradiente, aplicadas a imagens adquiridas e laboratórios de microscopia [58]. Dos Anjos et al. apresentaram um modelo *Watershed* que detecta manchas relacionadas com a localização de proteínas em imagens de gel eletroforese de duas dimensões [83]. Depois de uma validação automática, o método *Watershed* é utilizado de forma a “encher” a superfície do gradiente da imagem, a partir do centro das manchas. Este procedimento é influenciado por uma aplicação prévia do método *Watershed* à imagem original. Este modelo apresenta resultados bastante satisfatórios no processo de detecção das manchas, apresentando um número reduzido de falsos positivos e uma melhoria em termos de volume de processamento exigido pelo algoritmo.

Hodneland et al. realizaram alguns estudos relativos a um processo de comparação entre uma técnica *Watershed* e uma técnica *level-set* em imagens celulares de fluorescência [68][69]. Ambas as técnicas são largamente influenciadas por um sistema de aquisição de marcas. Esta dependência é uma consequência da localização e da dimensão das marcas. O modelo *Watershed* assume-se como tendo um desempenho mais efetivo, em comparação com o modelo *level-set*. O modelo *Watershed* não é significativamente influenciado por variações em termos dos contornos dos segmentos, o que também é sustentado por uma adequada aplicação de um sistema de pré-processamento. Uma importante vantagem do método *level-set* relativamente ao método *Watershed* prende-se com as propriedades de alisamento da segmentação resultante. Os resultados incluídos neste artigo mostram também que é possível combinar os dois modelos, de forma a obter melhores resultados em termos de tempo computacional e melhor definição das imagens resultantes. Tse et al. aplicaram um modelo de segmentação robusto baseado nos modelos de *Watershed* e *level-set* [84]. Este modelo inclui um processo de extração *level-set* que determina as várias regiões contidas na estrutura celular. Esta técnica permite a distinção entre regiões de intensidade muito próxima em imagens de baixo contraste e também evita problemas de excessiva divisão celular. O modelo *Watershed* implementado é especialmente otimizado para processos de segmentação de regiões de elevada intensidade. Os resultados mostram que a combinação dos dois modelos apresenta resultados bastante efetivos, formando aplicações muito robustas e com elevados níveis de estabilidade.

3 Metodologia Implementada

Neste capítulo são descritos todos os procedimentos desenvolvidos neste projeto. O modelo de segmentação desenvolvido foi projetado, numa primeira fase, com o objetivo de ser aplicado a um conjunto de imagens microscópicas. Este conjunto de imagens é descrito de forma geral no **capítulo 3.1**. A esquematização do projeto, assim como as decisões relativas às técnicas a serem implementadas, foram realizadas tendo em conta várias características gerais das imagens em consideração. Dentro deste conjunto de características destacam-se:

- A elevada percentagem de ruído presente nas imagens;
- O reduzido contraste entre os objetos e a região do fundo das imagens;
- A elevada diferença de intensidades entre objetos presentes na mesma imagem e em regiões significativamente próximas entre si;
- A existência de várias colónias constituídas por um número elevado de elementos e a grande proximidade entre os mesmos;
- A grande variação em termos de forma e tamanho dos objetos a serem identificados.
- Elevada presença de pontos de saturação de intensidade, o que prejudica o processo de segmentação.

Neste sentido, houve uma especial preocupação em desenvolver um conjunto de procedimentos que tivessem a capacidade de ultrapassar, de forma eficaz, as diversas dificuldades levantadas pelas características já enunciadas, criando, desde modo, um modelo de segmentação que apresente um grau de eficiência bastante satisfatório. Para além disto, houve logicamente, numa segunda fase, a preocupação de desenvolver um algoritmo com uma considerável versatilidade, ou seja, que possa ser aplicado a imagens com diferentes propriedades ou adquiridas através de procedimentos laboratoriais distintos.

Os procedimentos desenvolvidos são aqui esquematizados em três passos:

1. Pré-processamento;
2. *Template matching*;
3. *Watershed*.

3.1 Imagens disponibilizadas

As imagens em estudo neste projeto foram adquiridas no laboratório *Laboratory of Biosystem Dynamics da Tampere University of Technology*. Estas imagens são apresentadas sob a forma de pequenos filmes, constituídos por 125 *frames* cada. Foram disponibilizadas cinco sequências de imagens em escala de cinzentos. Seguidamente é possível visualizar algumas dessas imagens.

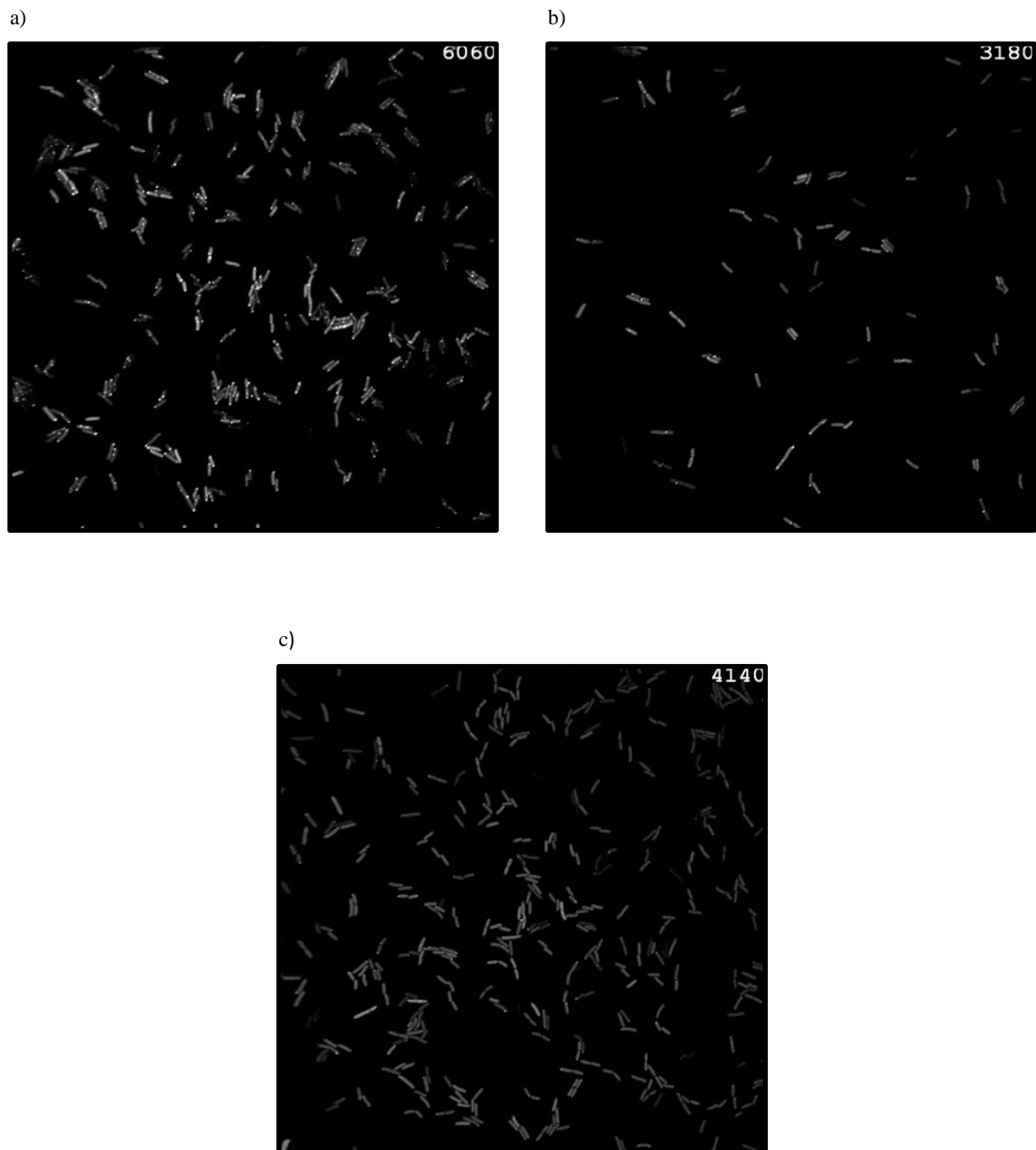


Figura 3.1: Exemplos de imagens disponibilizadas. a) Imagem 102 – Sequência 1; b) Imagem 54 – Sequência 2; Imagem 70 – Sequência 5.

3.2 Linguagem de programação C#

A linguagem C# assume-se como sendo uma linguagem moderna, enquadrada na programação orientada por objetos [85]. O C# é uma das linguagens inseridas na CLI (*Common Language Infrastructure*), uma estrutura desenvolvida pela Microsoft que estabelece o código e o tempo de execução que formam o núcleo da *Microsoft .NET Framework*, sendo uma linguagem que apresenta um *unified type system*, chamado de CTS (*Common Type System*). O CTS separa os *data types* em duas categorias: *Value types* e *Reference types*. Esta linguagem é tipicamente utilizada como suporte em softwares aplicados na área da engenharia, sendo base de softwares robustos e de elevada durabilidade e portabilidade. A mais recente versão do C# é a 5.0, libertada em 15 Agosto de 2012.

3.3 Biblioteca OpenCV

O OpenCV é designado como uma biblioteca *open source computer vision* [33]. Esta biblioteca foi criada de forma a aumentar a eficiência computacional em várias aplicações em tempo real. Inicialmente esta biblioteca foi implementada nas linguagens C e C++. O principal objetivo do OpenCV é fornecer uma estrutura simples que auxilie o utilizador a construir aplicações vision de forma rápida e com relativo grau de eficiência. Esta biblioteca contém mais de 500 funções que são úteis em variadíssimos campos como em imagem médica, user interface, sistemas de segurança, robótica, entre outros. OpenCV também contém uma sub-biblioteca baseada em *machine learning* chamada MLL (*Machine Learning Library*).

3.4 Pré-processamento

As técnicas de pré-processamento são de extrema importância na área do processamento de imagem, tal como já descrito no **capítulo 2.2**. O tipo de pré-processamento aplicado depende grandemente das características das imagens em estudo. Neste contexto, o primeiro grande objetivo ao realizar este projeto, centrou-se na aplicação de uma sequência de técnicas que permita um resultado satisfatório, tendo em conta um posterior processo de segmentação.

A sequência de técnicas implementadas neste projeto foi definida segundo as necessidades das imagens em consideração no mesmo. Estas técnicas serão detalhadamente descritas ao longo deste capítulo.

3.4.1 Técnica de alisamento da imagem (Filtro Mediana)

Esta técnica é largamente utilizada em processamento de imagem e apresenta propriedades muito úteis, tendo em conta as características das imagens em estudo neste projeto. As imagens apresentam uma quantidade de ruído considerável. A aplicação de um

filtro de mediana permite a atenuação desta quantidade de ruído, sem que haja uma perda de contraste por parte da imagem. Assim sendo, é aplicado um filtro com uma máscara 3x3.

3.4.2 Técnica de melhoramento de contraste (Transformada de Laplace)

Tal como já foi debatido no **capítulo 2.2**, o objetivo principal deste tipo de técnicas é o de transformar a imagem de forma a que seja mais fácil uma posterior extração das características da mesma. Neste sentido, a aplicação de um operador Laplaciano permite uma maior evidenciação dos contornos dos objetos em estudo, melhorando grandemente o nível de contraste apresentado nas imagens em consideração. No contexto onde estão inseridos os objetivos evocados nesta dissertação, a aplicação deste operador assume-se com sendo de extrema importância, tendo em conta que as imagens contêm colónias de bactérias que se apresentam muito próximas entre si, dificultando a distinção individual das mesmas e uma posterior identificação.

Neste projeto foi aplicado um algoritmo inserido na biblioteca *OpenCV* que aplica um operador Laplaciano com uma abertura de tamanho nove. A seguinte ilustração apresenta exemplos da aplicação deste operador.

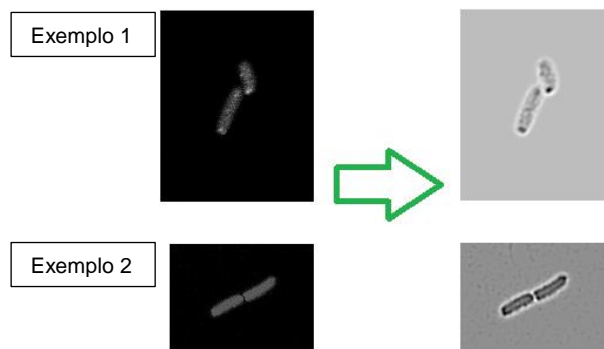


Figura 3.2: Exemplo de uma aplicação do operador Laplaciano implementado.

Seguidamente foi aplicado um procedimento que eleva ao valor máximo todos os pixels acima de um determinado nível de intensidade. Este nível é definido como sendo o valor da moda de todos os pixels da imagem, após a aplicação do operador laplaciano. Na seguinte ilustração é possível observar um exemplo da aplicação deste procedimento.

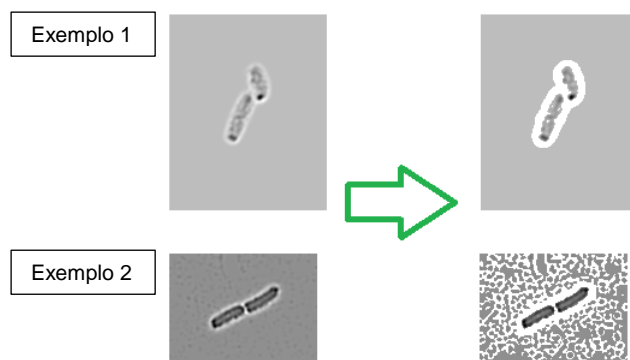


Figura 3.3: Exemplos da aplicação do procedimento de *threshold* aqui descrito.

3.4.3 Técnica morfológica (Operação de Abertura)

Tendo em conta que é de extrema importância um pré-processamento que procure isolar da melhor forma os objetos a serem segmentados, torna-se útil a utilização de uma transformação que, para além de permitir uma maior redução de ruído, permita uma maior definição das bactérias presentes na imagem. Neste sentido é aplicada uma operação de abertura em escala de cinzentos que tem em conta uma combinação entre um procedimento de erosão, seguido por um procedimento de dilatação.

3.4.4 Operação de remoção do ruído de fundo

Tal como foi já explicado, a remoção do ruído excessivo visualizado nas imagens é importante, tanto no sentido de um melhor processo de segmentação, como no sentido da necessidade de implementação de um algoritmo o menos pesado possível. Tendo isto em consideração, foi criado um procedimento de remoção completa do ruído presente no fundo das imagens em estudo.

Este procedimento é baseado num processo de binarização da imagem que permite a distinção entre o fundo e os grupos de bactérias presentes na imagem. Ao identificar a região pertencente ao fundo, facilmente é possível retirar todo o ruído presente nesta região. Seguidamente apresenta-se uma ilustração onde é possível verificar a execução do procedimento descrito.

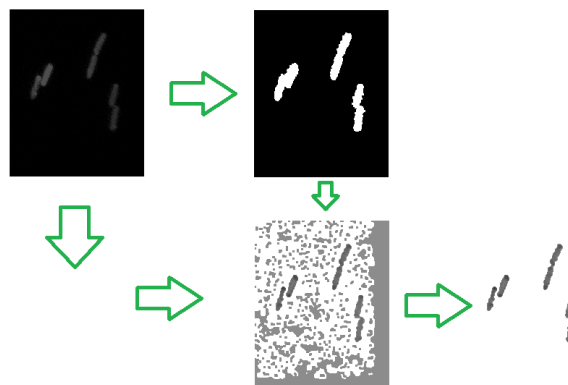


Figura 3.4: Exemplo do procedimento de remoção de ruído implementado.

3.4.5 Procedimento de redução dos pontos de elevada intensidade presentes nas bactérias

Um dos maiores problemas levantados ao longo deste projeto prende-se com a existência de vários pontos luminosos na grande maioria das bactérias presentes nas imagens fornecidas. Seguidamente apresentam-se duas das imagens fornecidas, na qual é possível visualizar a situação aqui descrita.

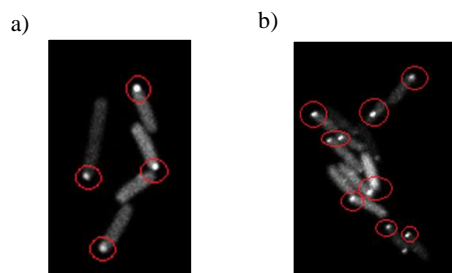


Figura 3.5: Dois dos exemplos que ilustram a evidência dos pontos de saturação presentes nas imagens.

Estas regiões correspondem a zonas de elevada saturação que, embora possam ser preponderantes no procedimento de vários estudos biológicos, relativos às imagens microscópicas em questão, prejudicam de forma severa todo o processo de segmentação. O resultado do operador Laplaciano descrito anteriormente quando aplicado às imagens em estudo, mostra que a presença das zonas de saturação faz com que estas sejam evidenciadas, ficando o objeto irremediavelmente “partido”, o que prejudica posteriormente os resultados adquiridos através do método *Template matching* e consequentemente, todo o processo de segmentação dos objetos. A figura seguinte apresenta um exemplo do processo aqui descrito.

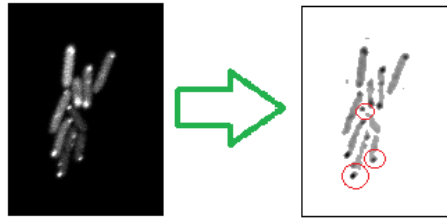


Figura 3.6: Exemplo das consequências da existência dos pontos de saturação após o pré-processamento.

De forma a atenuar a influência destas zonas de elevada saturação, foi definida uma sequência de procedimentos que, para além de reduzir estas mesmas zonas, melhora a qualidade das imagens em estudo. Esta sequência divide-se nos três passos seguidamente descritos.

Processo de reajustamento dos valores saturados

Este procedimento foi criado com o objetivo de diminuir a intensidade dos pixels mais saturados (correspondentes na sua grande maioria às zonas onde se situam os pontos luminosos). O processo de diminuição de intensidades não pode ser feito do mesmo modo em toda a imagem em estudo, tendo em conta as elevadas variações de intensidade que as bactérias presentes nas imagens apresentam. Este processo deve ser então realizado de uma forma mais zonal, de modo a ter em conta a localização das colónias de bactérias nas imagens em estudo. Assim sendo, é extremamente importante aplicar um procedimento que identifique os diferentes grupos de objetos e que posteriormente defina um valor de threshold específico e correspondente a cada grupo. Por outro lado, é importante que neste procedimento sejam definidos alguns parâmetros, de modo a evitar alterações prejudiciais em zonas que não correspondem às regiões de saturação.

Este procedimento pode ser explicitado através dos seguintes passos:

- Localização de cada uma das colónias: cada grupo de objetos é identificado através de um procedimento auxiliado por uma função incluída na biblioteca *OpenCV* denominada de *Find Contours*.
- Cálculo da média dos pixels correspondentes a cada grupo de objetos: apenas são contabilizados os pixels com uma intensidade entre 70 e 235, de forma a não selecionar a região do fundo e as zonas de saturação.
- Reposição dos valores dos pixels pelos novos valores médios: os valores médios guardados para cada grupo serão então os novos valores dos pixels correspondentes, desde que apresentem um valor inicial superior a 150.

Mediana entre imagens consecutivas

Tal como introduzido no **capítulo 3.1**, as imagens em estudo neste projeto estão organizadas em sequências temporais. De *frame* para *frame*, estas imagens possuem alterações pouco significativas, tendo em conta que cada filme apresenta um número bastante elevado de *frames*. Tendo isto em consideração, é possível executar pequenas operações entre imagens consecutivas, de forma a melhorar a sua qualidade.

Neste sentido, definiu-se a aplicação de um operador que executa a mediana entre cinco imagens consecutivas, ou seja, os valores correspondentes aos pixels de cada uma das cinco imagens e que se encontram numa determinada posição (x,y) são analisados, sendo determinado o valor mediano. Este será o novo valor do pixel na posição (x,y) da imagem resultante e este processo repete-se para todos os valores (x,y) . Seguidamente apresenta-se um exemplo da aplicação deste procedimento.

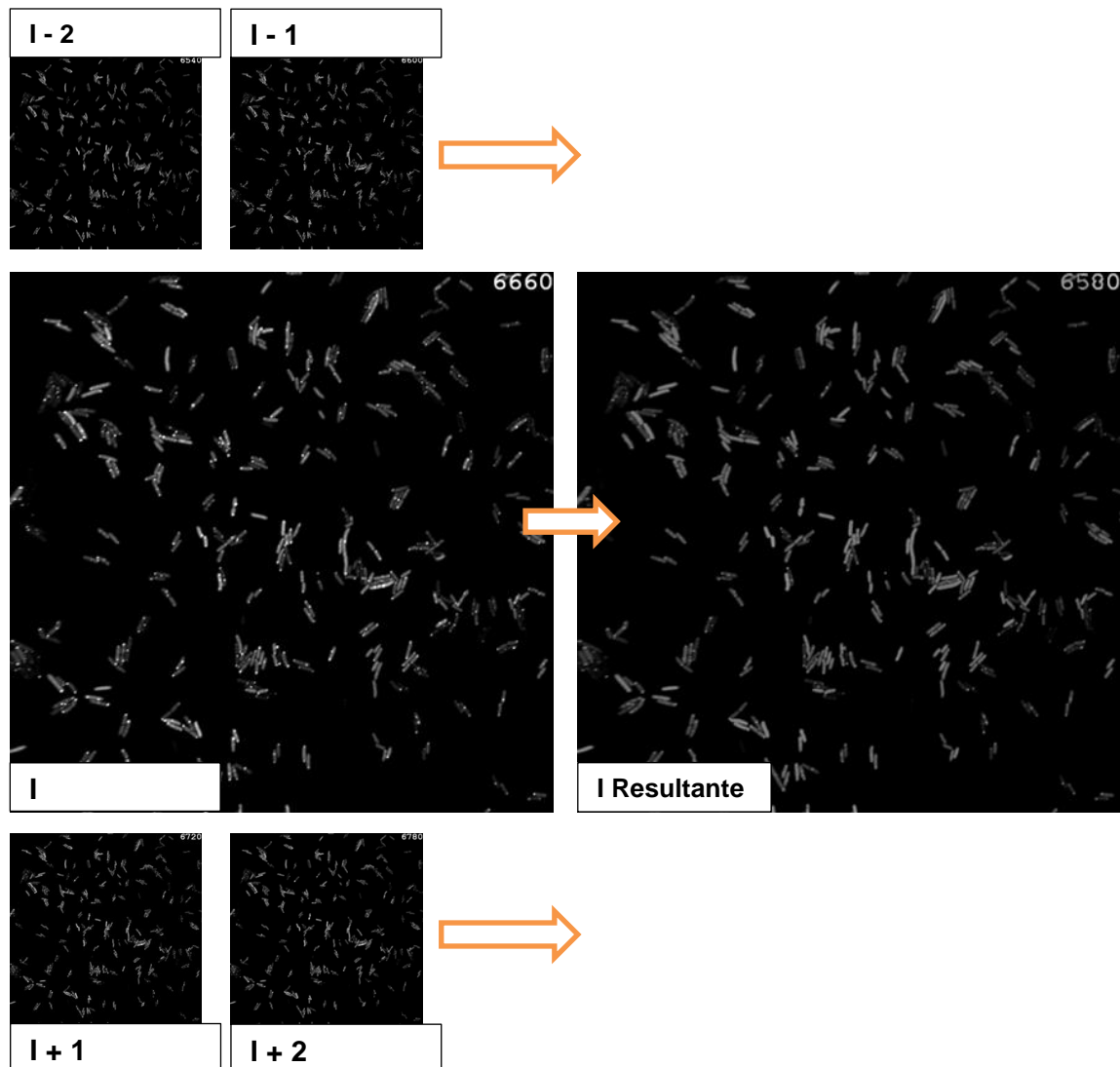


Figura 3.7: Aplicação do processo de mediana para a imagem I, tendo em conta as duas imagens anteriores e as duas imagens posteriores.

Mínimo entre imagens consecutivas

Este procedimento é semelhante ao descrito a cima, mas em vez de ser escolhido o valor mediano, é escolhido o valor de menor intensidade. O processo aqui descrito é executado logo após o processo de mediana e revela-se extremamente útil na remoção dos pontos luminosos. Seguidamente apresenta-se um exemplo da aplicação deste procedimento.

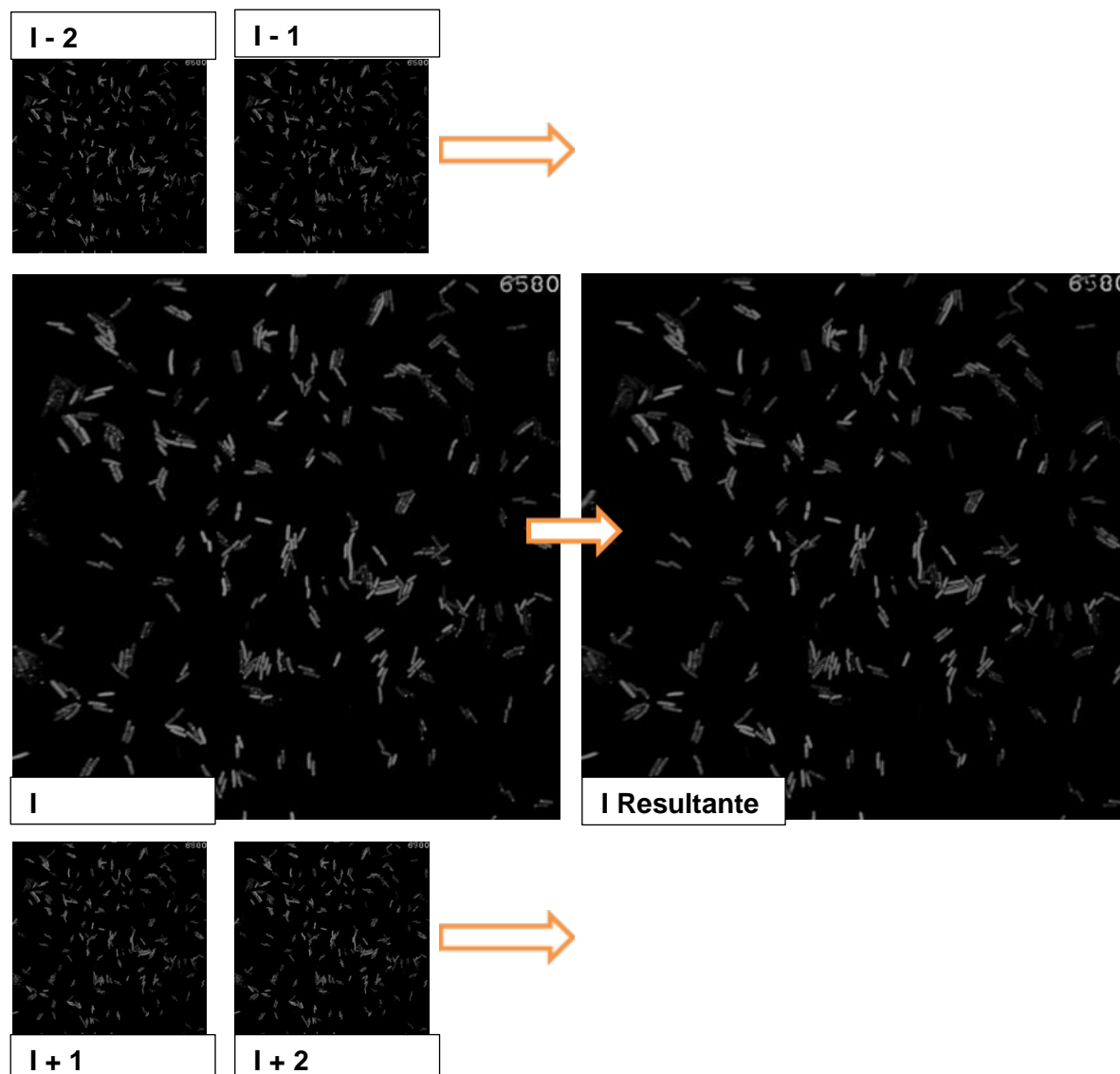


Figura 3.8: Aplicação do processo de mínimo para a imagem I, tendo em conta as duas imagens anteriores e as duas imagens posteriores.

A sequência de procedimentos aqui definida apresenta os seguintes pontos como principais vantagens:

- Redução das zonas de saturação;

- Redução da percentagem de ruído, devido essencialmente ao procedimento relativo à mediana da sequência de imagens;
- Redução da variação de intensidades entre objetos;
- Capacidade de manter as regiões de elevado contraste entre objetos.

Todos os métodos de pré-processamento implementados neste projeto e que foram aqui descritos são esquematizados no seguinte fluxograma.

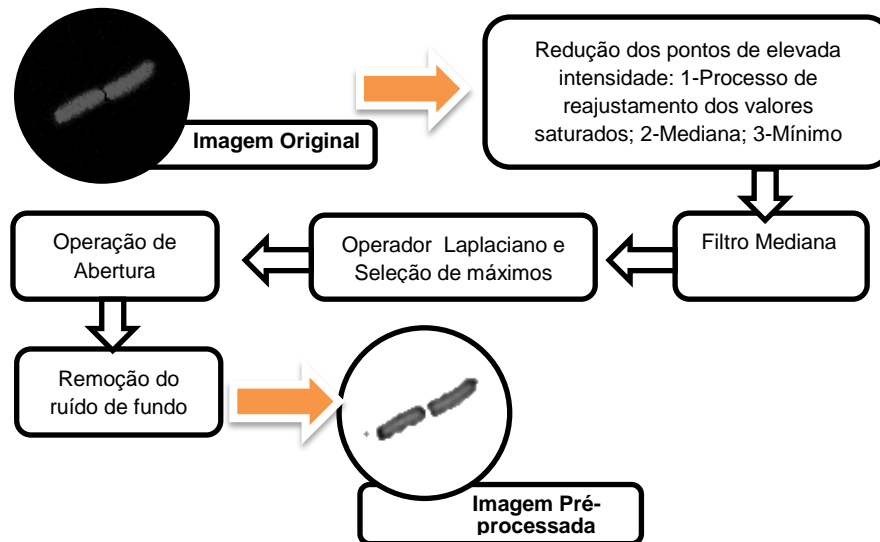


Figura 3.9: Esquema resumido do conjunto de procedimentos de Pré-processamento.

3.5 Template matching

Seguidamente ao pré-processamento é aplicado um procedimento baseado em *Template matching*. Este método é extensamente descrito no **capítulo 2.3**. A aplicação deste método é realizada tendo em conta a elevada complexidade das imagens em estudo. As imagens apresentam um ruído considerável e um nível de contraste relativamente baixo, o que, aliado à variação de intensidades visualizada em cada um dos pixels constituintes, torna todo o processo de segmentação da imagem mais lento. O resultado proveniente do pré-processamento é já uma boa aproximação àquilo que se pretende de forma a adquirir as marcas necessárias à aplicação do modelo *Watershed*. Mas este procedimento por si só mostra-se insuficiente para um futuro processo de obtenção de marcas para cada bactéria presente na imagem. Numa breve visualização das sequências de imagens em estudo neste projeto, facilmente visualizam-se colónias de bactérias que apresentam objetos a serem detectados, que se encontram demasiadamente próximos entre si, possibilitando assim a existência de sub-segmentos. Seguidamente apresentam-se vários exemplos que ilustram esta situação.

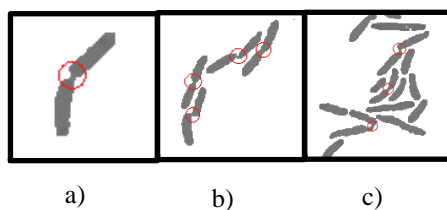


Figura 3.10: Exemplos de situações de união entre objetos a serem segmentados.

A correcção destes sub-segmentos não apresenta resultados satisfatórios com uma simples aplicação de técnicas morfológicas, podendo originar a geração de segmentos que não existem na realidade. Para contornar este problema, a aplicação de um modelo baseado em *Template matching* torna-se uma boa solução, tendo em conta as propriedades fundamentais deste método. A ideia chave desta aplicação é no fundo a detecção da localização dos vários objetos a serem identificados, através da comparação de cada região da imagem com uma sequência de *templates* pré-definida. Este processo de comparação tem como resultado a criação de um mapa de intensidades que quantifica os pontos da imagem em estudo que apresentam mais semelhança com cada um dos *templates*. Neste mapa é definido um determinado nível de threshold que posteriormente corresponderá então à definição das regiões da imagem sobre as quais serão geradas as diferentes marcas correspondentes a cada bactéria. Na ilustração apresentada a baixo é possível visualizar um exemplo que permite uma melhor percepção do procedimento aqui descrito.

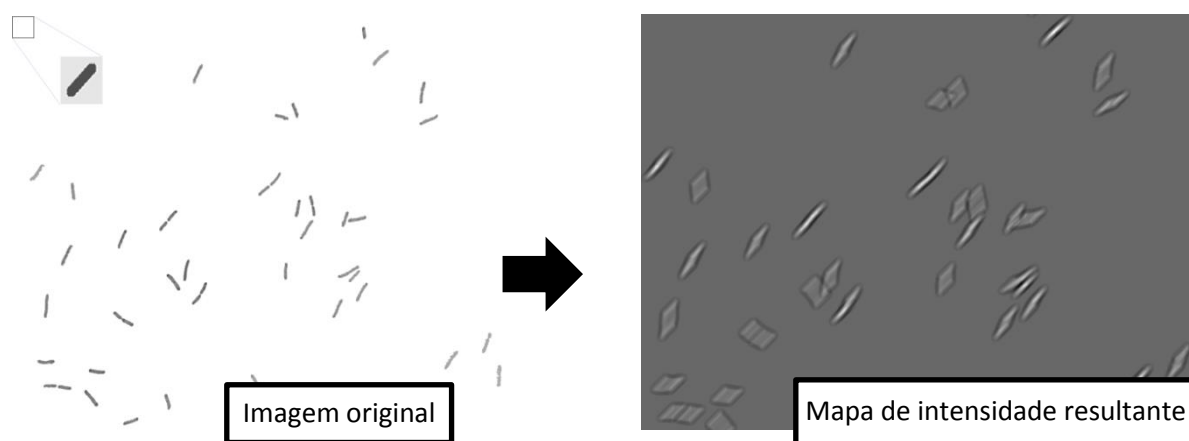


Figura 3.11: Exemplo de um mapa de intensidades resultante do processo implementado.

Ao longo desta secção serão descritos todos os passos do algoritmo de *Template matching* implementado neste projeto.

3.5.1 Implementação geral

O método aqui proposto é invariante em termos de escala, rotação e intensidade. A técnica implementada consiste em determinar a correspondência entre cada um dos *templates* pré-definidos com os objetos de cada uma das imagens em estudo. Através do algoritmo seguidamente descrito, é analisada cada imagem, comparando pequenas secções da mesma com o *template*. Estas secções apresentam as mesmas dimensões do *template* definido e podem ser vistas de uma forma mais intuitiva como uma janela que a cada aplicação do algoritmo de correlação, move-se de forma a percorrer toda a imagem. Tal como se pode observar na seguinte ilustração, a janela desloca-se sempre uma coluna para a direita até atingir o comprimento da imagem em questão. Ao atingir este ponto, desloca-se uma linha para baixo e volta a deslocar-se coluna a coluna, “varrendo” toda a imagem. Tendo isto em conta, a imagem resultante deste processo terá as seguintes dimensões:

$$[R(W); R(H)]: [I(W) - T(W) + 1; I(H) - T(H) + 1]$$

Em que R define a imagem resultante deste processo, I define a imagem em estudo, T define o *template*. W e H definem, respetivamente, a largura e a altura de cada elemento. A seguinte imagem exemplifica este procedimento.

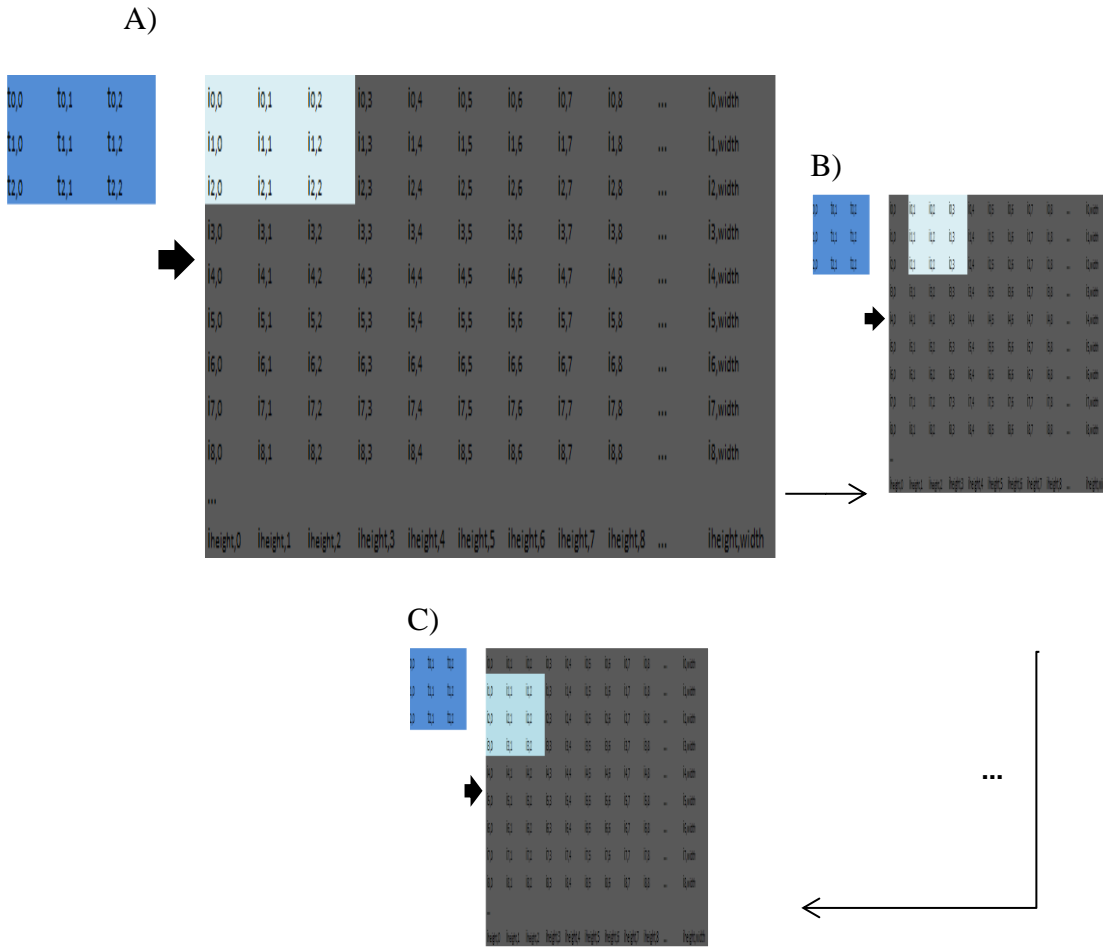


Figura 3.12: Esquema ilustrativo do processo de translação do *template* sobre a imagem.

Tendo em conta o processo aqui descrito, facilmente entende-se que um objeto, que se encontre muito próximo de uma das extremidades da imagem, não é devidamente correlacionado com o *template*. De forma a otimizar este processo (embora estas situações sejam muito pouco prováveis), o tamanho da imagem em estudo é redimensionado. Deste modo, a imagem passa a ter as seguintes dimensões:

$$[I'(W); I'(H)]: [I(W) + T(W) - 1; I(H) + T(H) - 1]$$

Em que I' é a imagem que resulta deste dimensionamento e T é designado o *template* de maiores dimensões contido na sequência de *templates* definida no processo.

O método *Template matching* aplicado neste projeto tem por base um algoritmo de correlação denominado de *Correlation coefficient matching*, definido da seguinte forma:

$$T'(x', y') = T(x', y') - (w \cdot h) \frac{1}{\sum_{x'', y''} T(x'', y'')}$$

$$I'(x + x', y + y') = I(x + x', y + y') - (w \cdot h) \frac{1}{\sum_{x'', y''} I(x + x'', y + y'')}$$

$$R_{ccoeff}(x, y) = \sum_{x', y'} [T'(x', y') \cdot I'(x + x', y + y')]$$

Em que $R_{ccoeff}(x, y)$ representa os novos valores que constituem a imagem resultante, I a imagem inicial, T o *template*, w e h representam, respetivamente, a largura e a altura do *template* e (x, y) representam a localização dos pixels dos elementos considerados.

Tal como se pode comprovar através das formulas descritas acima, é aqui apresentado um algoritmo matematicamente simples, que quando aplicado neste sistema, mostra ter resultados bastante satisfatórios. Resumidamente, é calculada a média dos pixels do *template*, assim como a média dos pixels da imagem que estão incluídos na janela selectiva descrita acima. Estes valores são subtraídos a cada pixel correspondente, ou seja, tanto os pixels do *template* como os pixels da janela selectiva são alterados. Seguidamente e para cada pixel, é calculada a multiplicação entre os novos valores do *template* e os novos valores da janela selectiva. Este valores são posteriormente somados e o valor resultante deste somatório é designado como o valor do pixel correspondente na imagem resultante. Este processo repete-se para cada translação da janela selectiva.

Com a implementação deste algoritmo sobre a imagem em estudo, é gerada uma imagem resultante que corresponde a um mapa de intensidades (exemplificado na **Figura 3.11**). As regiões com intensidades mais elevadas (máximos locais) são as zonas onde existe uma maior correlação entre a imagem e o *template* [86]. Este mapa é posteriormente limitado aos níveis máximos de intensidade através da aplicação de um nível de threshold pré-definido. Este procedimento repete-se em cada processo de rotação do *template*. Este processo de rotação é descrito ao pormenor mais à frente.

De forma a seleccionar as regiões onde há uma maior correlação entre a imagem e os *templates*, ou seja, as regiões correspondentes aos diferentes máximos locais, é necessário definir um determinado threshold, ou seja, um valor específico que irá limitar estes mesmos máximos. A escolha deste valor não é tão linear quanto possa parecer. Existem várias formas distintas de proceder a esta definição. Deste modo, estipulou-se como método base o cálculo da razão entre o valor de cada pixel da imagem resultante e o valor individual resultante do processo de correlação do *template* por ele próprio (Esta razão é descrita na equação abaixo). O valor da correlação do *template* por ele próprio é considerado o valor máximo de correlação que pode ser obtido para um *template* específico, ou seja, a razão calculada terá obrigatoriamente de ser igual ou inferior a 1. O valor máximo de correlação é repetidamente calculado de cada vez que determinado *template* é rodado e este processo repete-se para cada um dos *templates* utilizados.

$$R'(x, y) = \frac{R(x, y)}{M}$$

Em que R designa os valores dos pixels da imagem resultante e M o valor máximo de correlação. Os valores resultantes da razão calculada encontram-se então entre 0 e 1.

Posteriormente é definido um nível limite (denominado de $\mu_{template}$) que selecciona os máximos locais, tal como esquematizado seguidamente.

$$R''(x, y) = \begin{cases} 1, & \text{se } R'(x, y) \geq \mu_{template} \\ 0, & \text{caso contrário} \end{cases}$$

Em que $R''(x, y)$ define todos os valores que foram seleccionados por esta condição. O termo $\mu_{template}$ representa então o limite definido de forma a seleccionar os valores correspondentes aos máximos locais da imagem resultante e terá sempre esta conotação ao longo desta dissertação.

3.5.2 Definição dos *templates* gerais

Os *templates* utilizados no processo de *Template matching* foram definidos com base nas características das bactérias presentes na imagem. Deste modo foram tidas em conta propriedades como a forma, o tamanho e a intensidade dos objetos presentes nas imagens em estudo. Tendo em conta as elevadas variações das bactérias em estudo, relativamente às suas dimensões, é definida uma sequência de cinco *templates*, com o intuito de que a detecção das bactérias seja feita de forma mais eficiente. Um dos grandes problemas da aplicação de *templates* em processos de *Template matching* é o facto de geralmente existirem significativas variações em termos de sensibilidade, relativamente à forma dos objetos presentes na imagem ou às diferenças de iluminação. Um dos principais objetivos inseridos no projeto realizado foi o de atenuar estas mesmas variações.

Seguidamente será apresentada a sequência dos *templates* utilizados neste projeto.

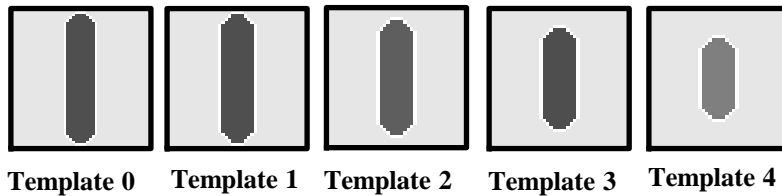


Figura 3.13: Sequência de *templates* definida.

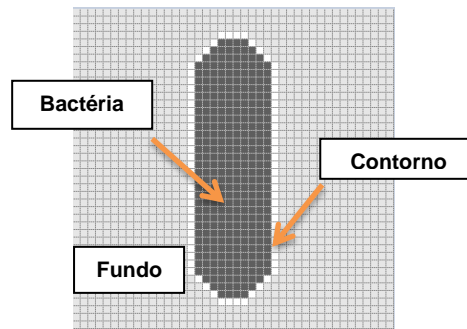


Figura 3.14: Esquema ilustrativo de um dos *templates*.

Tal como se pode verificar na **Figura 3.13**, os cinco *templates* aqui definidos apresentam variações relativamente ao tamanho do objeto a ser detectado. Estas definições foram instituídas tendo por base o tamanho das bactérias presentes nas imagens em estudo. Através da ilustração é possível também verificar que houve uma especial preocupação em relação à distinção entre o fundo dos *templates* e a zona representativa do objeto a detectar (**Figura 3.14**). Repare-se que o facto de existirem várias colónias de células no tipo de imagens apresentado, faz com que a distância espacial entre estas células seja muito reduzida. Assim sendo, tendo em conta o processo de correlação aqui implementado, a combinação de um determinado objeto presente na imagem com o respectivo *template* poderá ser altamente influenciado pelos objetos que o rodeiam, tornando a combinação muito menos precisa e alterando grandemente os resultados de todo o processo. Tendo todos estes factos em consideração, torna-se então necessário restringir o processo de correlação apenas à zona do *template* correspondente ao objeto, ou seja, os pixels da região de fundo do *template* não são correlacionados com a imagem. Esta situação claramente torna todo o processo de *Template matching* mais eficiente e robusto.

Seguidamente são ilustrados alguns exemplos das aplicações de cada *template* sobre uma imagem pré-definida. O valor de $\mu_{template}$ definido é especificado na legenda de cada figura.

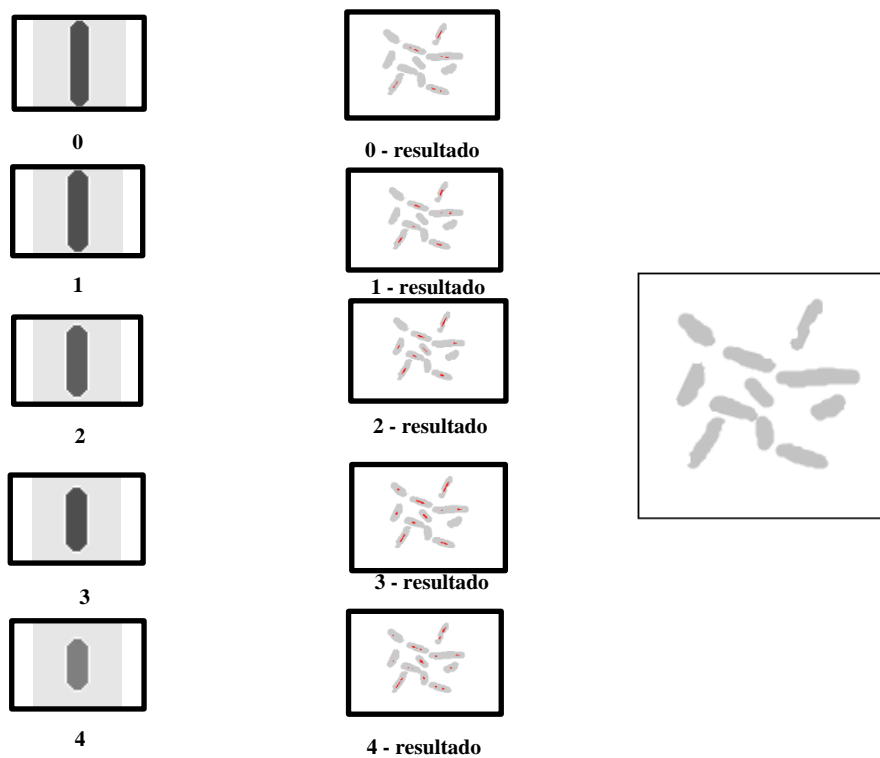
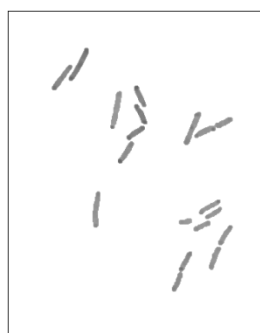


Figura 3.15: Exemplos das aplicações de cada *template* sobre uma imagem sintética ($\mu_{template}=0,75$).



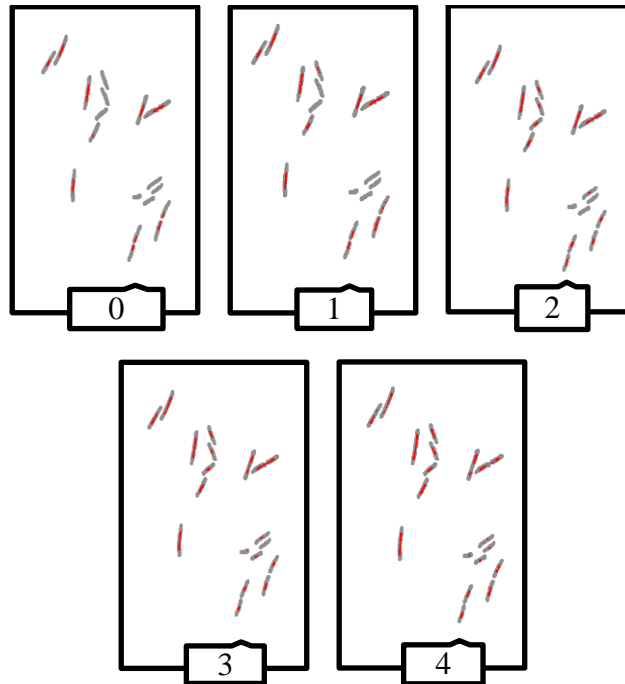


Figura 3.16: : Exemplos das aplicações de cada *template* sobre uma imagem real e pré-definida ($\mu_{template}=0,60$).

Tendo em conta as diferentes orientações dos objetos, torna-se incrivelmente vital a criação de um modelo *Template matching* que, para além de invariante em termos de escalamento, seja invariante em termos de rotação. Neste sentido, foi criado um procedimento de rotação para cada um dos *templates*. Cada processo de rotação apresenta uma variação angular de 5° . Este valor foi definido no sentido de tentar implementar um sistema que detectasse cada objeto, independentemente da orientação angular deste e ao mesmo tempo, que fosse o menos pesado possível a nível computacional. A seguinte ilustração exemplifica este processo.



Figura 3.17: Representação dos vários *templates* aplicados, mediante o processo de rotação implementado.

3.5.3 Definição de um *template* específico de detecção das extremidades das bactérias

Ao longo da fase de testes que foi aplicada ao processo de *Template matching* implementado neste projeto e através da análise das imagens resultantes, sentiu-se a necessidade de definir um *template* adicional e diferente da sequência já definida. A imagem seguidamente apresentada é um bom exemplo deste facto. Repare-se que muitas vezes, ao visualizar as diferentes imagens em estudo, verifica-se que as diferentes células acabam por ligar-se umas às outras, estabelecendo pontos de união que muitas vezes parecem fazer parte da própria estrutura celular. O próprio algoritmo de correlação implementado associa em muitos casos duas ou mais células que, visualmente, devem ser identificadas de forma individual. Tendo isto em consideração, o *template* aqui descrito tem por objetivo evitar situações de sub-segmentação. A ideia geral da criação deste *template* é a identificação de extremidades de objetos individuais que se encontram unidos (ver **Figura 3.20– B) Extremidade**). Este é sempre o primeiro *template* a ser aplicado à imagem. Os máximos locais selecionados a partir deste *template* são localizados e estes pontos, após sofrerem um processo de dilatação, não são contabilizados no posterior processo de identificação realizado com os restantes *templates*. Este *template*, bem como alguns dos resultados da sua aplicação, são ilustrados seguidamente.

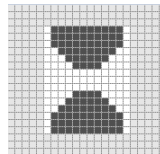


Figura 3.18: *Template* utilizado no processo de detecção de extremidades de bactérias.

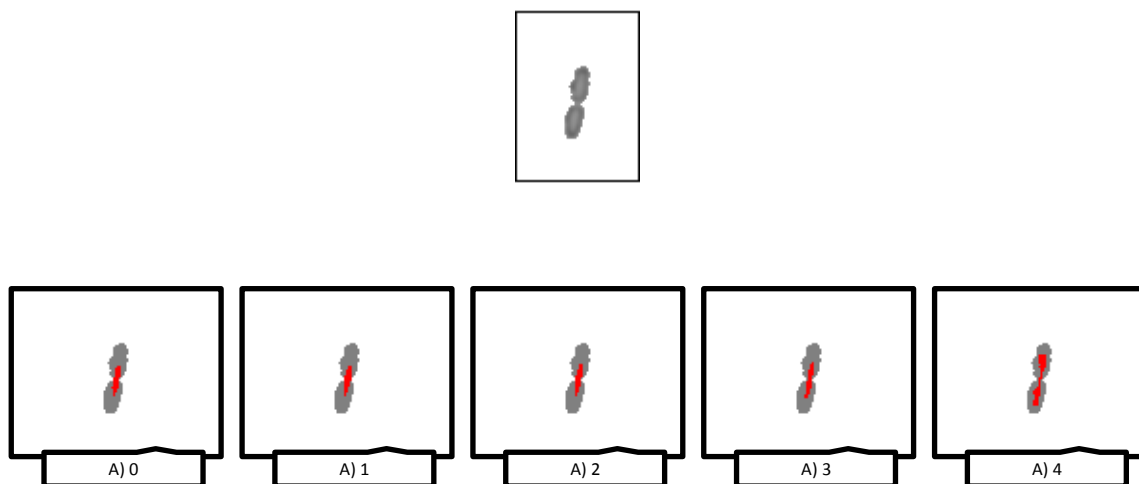


Figura 3.19: Aplicação do procedimento de detecção de bactérias sem ter em conta a detecção de extremidades ($\mu_{template}=0,60$).

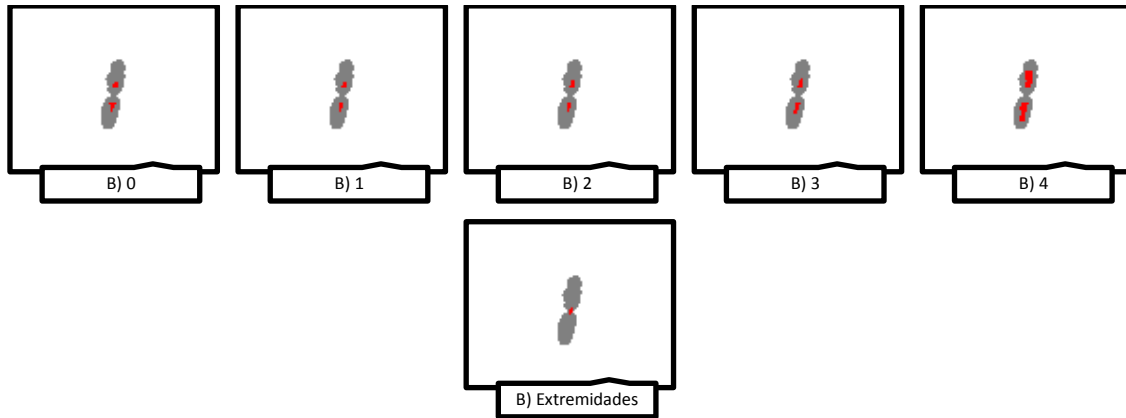


Figura 3.20: Aplicação do procedimento de detecção de bactérias tendo em conta o processo de detecção de extremidades ($\mu_{template}=0,60$ e $\mu_{extremidade}=0,68$).

As Figuras 3.19 e 3.20 ilustram um exemplo de duas bactérias unidas que só são identificadas com as respectivas marcas como duas bactérias individuais através do procedimento aqui descrito (ver Figura 3.21).

O parâmetro $\mu_{template}$ não é aplicado a este *template*, tendo em conta que este apresenta um grau de sensibilidade mais elevado quando aplicado à imagem em estudo. Assim é definido um novo parâmetro denominado de $\mu_{extremidade}$, que apresenta o mesmo significado que o $\mu_{template}$ mas apenas aplicado ao *template* de extremidades.

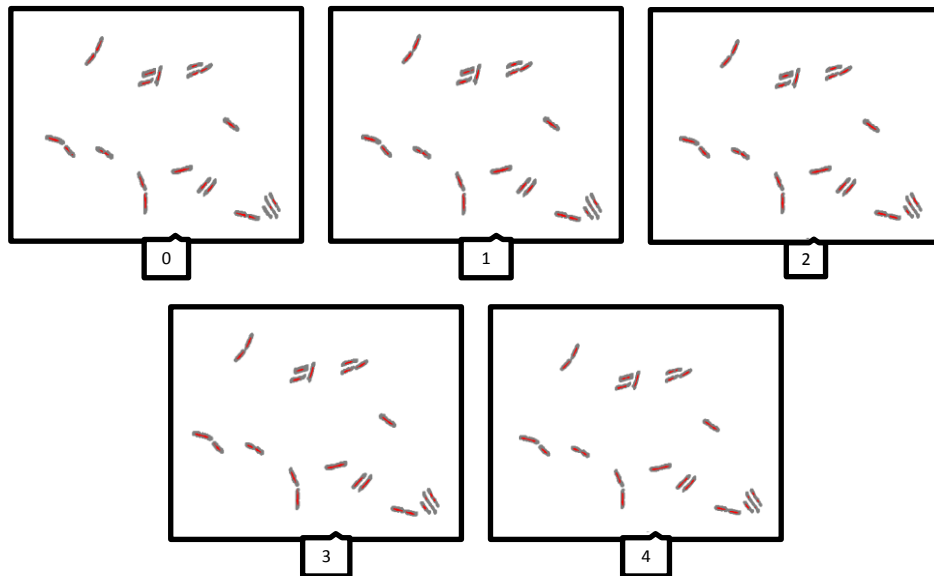




Figura 3.21: Exemplo da aplicação do *template* aqui descrito.

3.5.4 *Template matching*- Esquematização final

Tal como foi já referido, o procedimento base aqui implementado repete-se de cada vez que é aplicada uma rotação ao *template*. Este processo, por sua vez, é repetido para toda a sequência de *templates*, incluindo o *template* de detecção de extremidades. Por cada procedimento é gerada uma imagem resultante que representa os máximos locais, indicando as localizações das zonas de maior correlação entre a imagem em estudo e o *template*. Estes máximos são seleccionados através do processo já descrito. Estes valores são seguidamente guardados numa matriz. Esta matriz tem como principal objetivo acumular os valores obtidos em cada procedimento de rotação do respectivo *template*. Após a finalização de todos esses processos, é gerada uma imagem em que são visualizados os objetos a serem detetados, assim como as diferentes marcas situadas nos pontos correspondentes aos máximos locais. Este processo é feito com o auxílio da matriz de valores já descrita. Este processo repete-se para todos os restantes *templates*. Deste modo, são geradas cinco imagens correspondentes aos cinco *templates* que foram utilizados. Adicionalmente é gerada uma imagem para o *template* de detecção de extremidades.

O procedimento de *Template matching* implementado neste trabalho é finalizado com a geração de uma imagem final, constituída pela soma das marcas incluídas em cada uma das imagens resultantes de cada um dos *templates*. Nesta imagem final é possível visualizar as marcas geradas para cada objeto, marcas estas que serão utilizadas posteriormente no processo de *Watershed*, de forma a proceder à segmentação da imagem. O seguinte fluxograma resume todo o processo de *Template matching* aqui descrito.

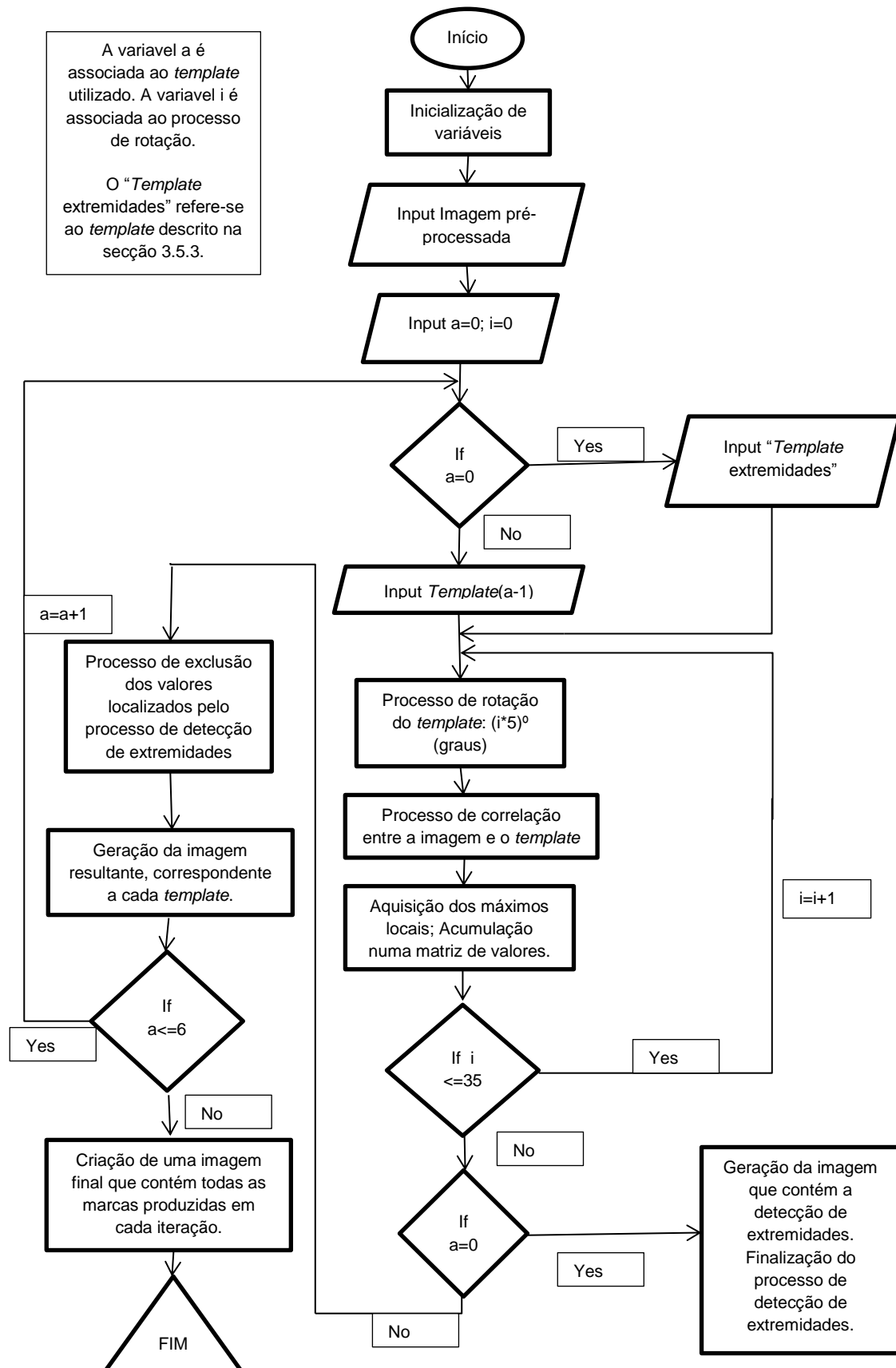


Figura 3.22: Fluxograma geral do processo de *Template matching* implementado.

3.6 *Watershed*

O método de *Watershed* constitui o passo final no processo de segmentação proposto neste trabalho. A segmentação dos objetos presentes nas imagens em estudo tem por base um método *Watershed* com recurso a marcas específicas. Estas marcas correspondem aos mínimos locais presentes nas imagens em estudo, indicando, conseqüentemente, a localização dos vários objetos a serem detetados. As marcas necessárias são obtidas através do procedimento de *Template matching* previamente executado (descrito no **capítulo 3.3**). O método de segmentação *Watershed* é descrito no **capítulo 2.4**, sendo um dos modelos mais utilizados no campo da segmentação de imagem.

O procedimento implementado neste projeto é iniciado com a aquisição das marcas previamente geradas. Estas marcas sofrem posteriormente um processamento simples, realizado com objetivo de melhorar a eficiência do algoritmo em termos de resultados finais. Seguidamente serão descritos os passos essenciais deste processamento.

3.6.1 Aquisição e tratamento das marcas geradas

Tal como foi referido no **capítulo 3.3**, Uma das imagens geradas a partir do procedimento de *Template matching*, anteriormente executado, contem as marcas obtidas a partir do *template* de detecção de extremidades. Estas marcas serão úteis no processamento das marcas gerais, tal como se verá mais à frente. De forma a facilitar a compreensão do processo, estas marcas serão denominadas de “marcas de extremidade”.

3.6.2 Dilatação das marcas gerais

Inicialmente as marcas sofrem uma operação de dilatação com duas iterações. Esta operação é realizada segundo determinadas condições. Neste ponto existe a preocupação de que as diferentes marcas não se unem entre si, ou que não ultrapassem os limites impostos pelos contornos das bactérias. Assim sendo, o processo de dilatação é realizado apenas para os pixels que não pertencem ao fundo da imagem e para os pixels em que a intersecção destes com as regiões onde se situam as marcas de extremidades seja nula. Seguidamente é ilustrado um exemplo que permite uma melhor compreensão deste processo.

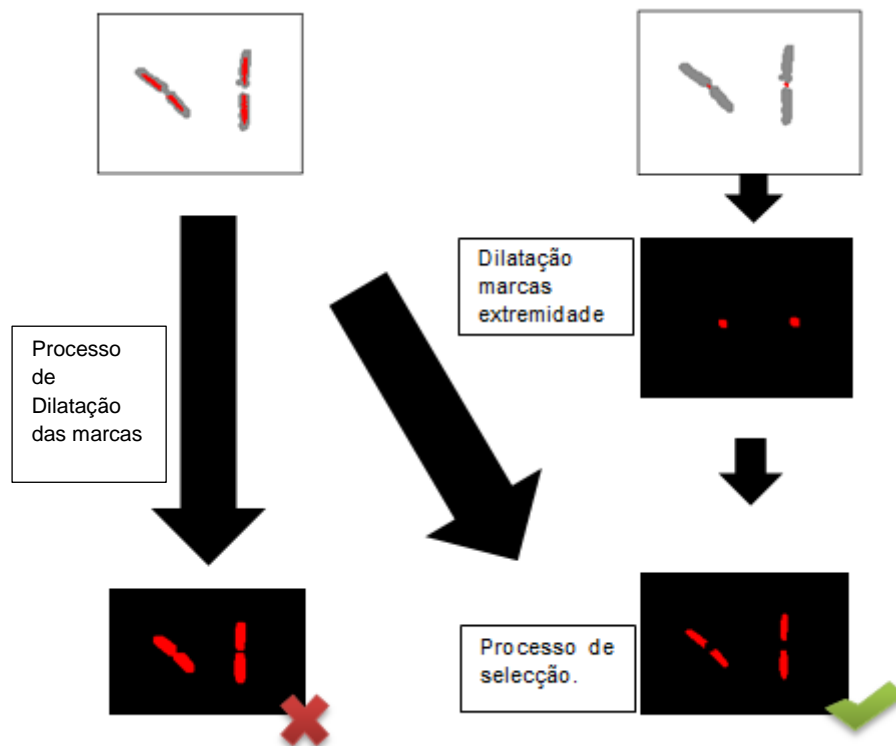


Figura 3.23: Exemplificação do processo de tratamento de marcas aqui descrito.

3.6.3 Contagem de marcas

Após o processo de dilatação das marcas, é executado um procedimento de contagem das mesmas. Esta contagem é essencial para o posterior processo de segmentação a partir de um método de *Watershed* baseado em marcas. Esta contagem é feita com o auxílio do procedimento *Find Contours* incluído na biblioteca *OpenCV*. Este procedimento identifica os contornos de cada uma das marcas, estabelecendo um processo de contagem e etiquetação de cada objeto.

3.6.4 Algoritmo *Watershed*

O algoritmo aqui implementado recebe como argumento o conjunto de marcas anteriormente processadas. Após a execução de um procedimento que gera uma marca correspondente à região do fundo da imagem, etiquetada com o valor 1, o conjunto de marcas é então etiquetado segundo a seguinte expressão:

$$m(i) = 2; 3; 4; \dots; i + 1$$

Em que m representa o número etiquetado e i o número de marcas pertencentes ao conjunto. Seguidamente o algoritmo *Watershed* é executado de forma a segmentar a imagem

em estudo. As regiões localizadas pelas diferentes marcas são rigorosamente reconhecidas e conetadas com os mínimos locais produzidos.

Seguidamente é executado um procedimento que exclui todos os segmentos com uma dimensão inferior a um valor previamente definido. Decidiu-se a exclusão de todos os segmentos com um número de pixels inferior a 70. Isto permite que não sejam contabilizados segmentos de pequenas dimensões que não correspondem a bactérias. A seguinte imagem exemplifica este procedimento.

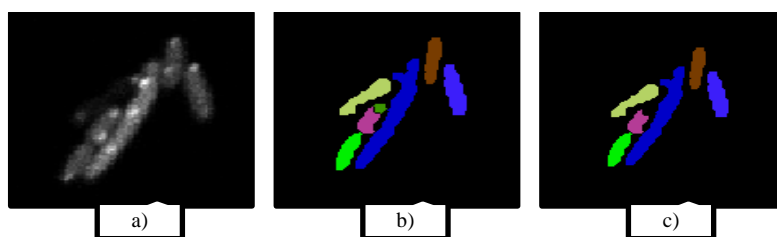


Figura 3.24: Exemplo do resultado da exclusão de segmentos de dimensões reduzidas – a) imagem de entrada; b) imagem segmentada; c) imagem após a aplicação do processo de remoção de segmentos de pequenas dimensões.

Na figura seguinte é possível observar um exemplo de todo o processo aqui descrito.

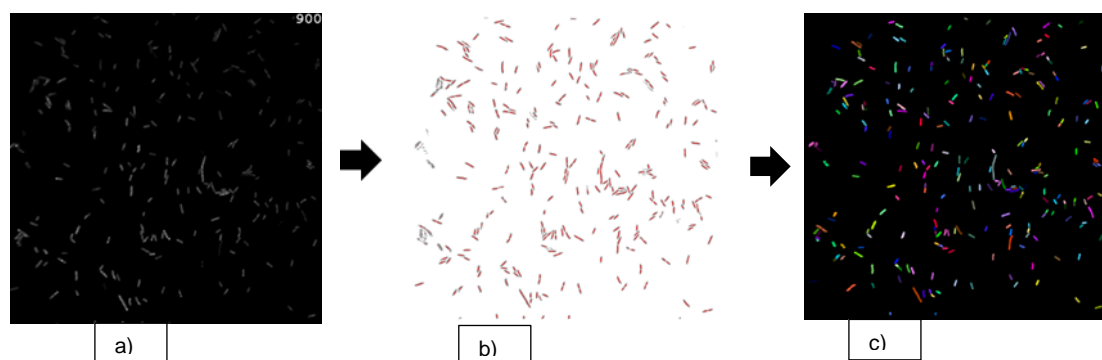


Figura 3.25: Exemplificação do resultado obtido através do processo de segmentação implementado – a) imagem de entrada; b) imagem resultante do processo *Template matching* com as marcas correspondentes; c) processo de segmentação através do método *Watershed*.

Após a geração das imagens devidamente segmentadas é necessário proceder à contagem automática do número de bactérias. Este processo é feito por intermédio da contagem de marcas, descrita anteriormente sendo finalizado com a geração de um documento em formato texto (.txt) que apresenta a lista de valores correspondente às contagens realizadas para cada uma das imagens em estudo. Na figura seguinte é possível visualizar um exemplo deste processo.

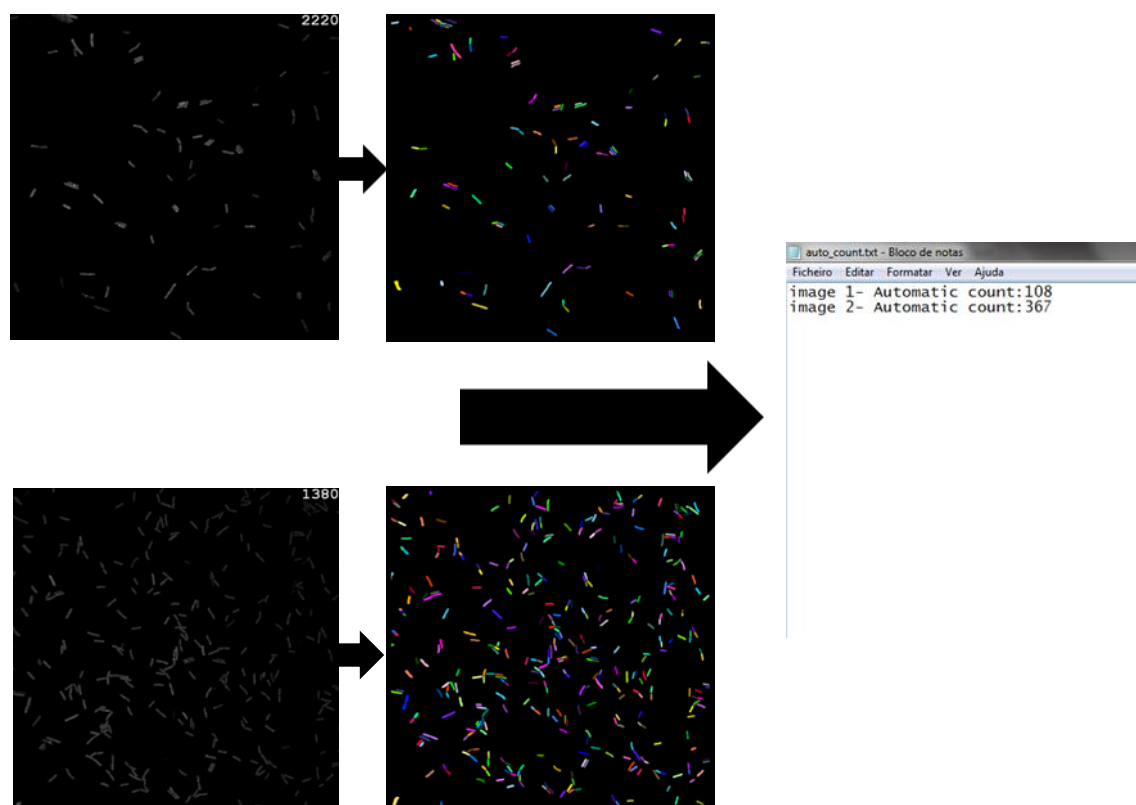


Figura 3.26: Exemplificação do procedimento de aquisição automática de um ficheiro contendo o número de bactérias presente nas imagens em estudo.

Seguidamente, todo o procedimento aqui descrito é esquematizado num fluxograma.

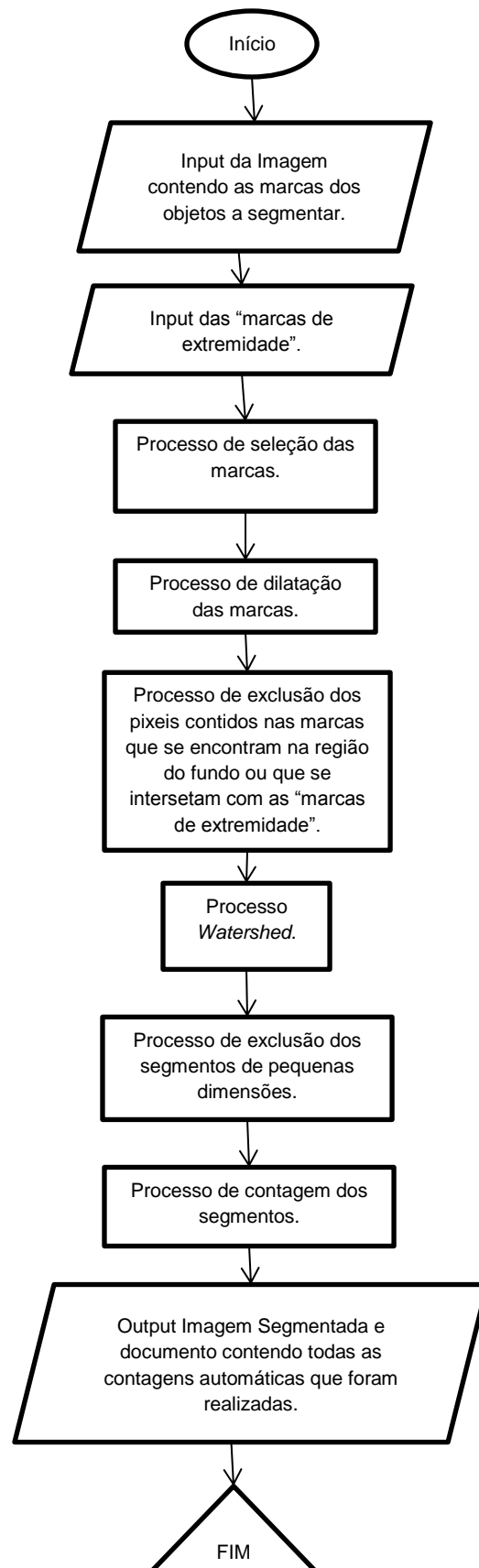


Figura 3.27: Fluxograma do procedimento *Watershed* implementado neste projeto.

3.7 Seguimento automático

Tal como enunciado na parte introdutória desta dissertação, um dos objetivos principais era o de desenvolver um procedimento de seguimento temporal do conjunto de amostras disponibilizadas. Neste sentido, foi desenvolvido um processo de seguimento automático que complementa o algoritmo de segmentação anteriormente descrito. De forma a que haja uma melhor compreensão dos procedimentos realizados, esta secção é dividida em três partes:

1. Processo geral de correção de segmentos;
2. Processo geral de seguimento automático;
3. Correção do processo de seguimento automático.

3.7.1 Processo geral de correção de segmentos

Para além de melhorar a eficiência do processo de segmentação do filme, evitando alguns casos de subssegmentação, este procedimento apresenta uma significativa importância no processo de *Tracking* efetuado à-posteriori, evitando que o seguimento temporal seja quebrado de *frame* para *frame*.

Inicialmente, é realizada uma correção dos segmentos resultantes do processo de segmentação que é seguidamente descrita. Considerando uma dada sequência de imagens Z , tal que $Z = \{Z_i; Z_{i+1}; Z_{i+2}; \dots; Z_{i+n}\}$ a imagem Z_{i+1} é corrigida através da comparação com a imagem anterior Z_i . O procedimento aqui implementado executa a interseção entre as duas imagens com o objetivo de verificar se um dado segmento é visualizado na imagem Z_i , mas não é visualizado na imagem Z_{i+1} , detectando assim uma incorreção. A detecção deste tipo de incorreções resolve grande parte dos erros de segmentação do passo anterior apresentando a vantagem de ser bastante simples e de rápida execução. Após todas as incorreções serem detectadas, a imagem Z_{i+1} é alterada de forma a que sejam incluídos os novos segmentos correspondentes a cada incorreção. Este processo repete-se para cada par de imagens consecutivas da sequência Z . A figura seguinte mostra um par de imagens nos quais se verifica uma incorreção deste tipo.

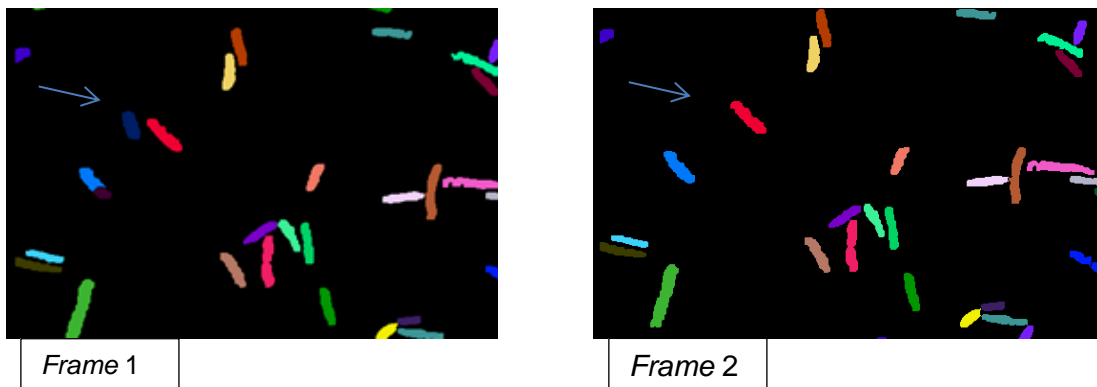


Figura 3.28: Ilustração exemplificativa de uma incorreção entre *frames* consecutivos.

3.7.2 Processo de seguimento automático

Tendo em consideração a sequência de imagens Z anteriormente descrita, efetua-se inicialmente a intersecção entre a imagem Z_i e a imagem seguinte, Z_{i+1} . O objetivo desta operação é determinar para cada segmento da imagem Z_{i+1} o seu correspondente em Z_i , ou seja, o segmento de Z_i que apresenta um maior nível de intersecção (intersecção com um maior número de pixels). A imagem Z_i é comparada pixel a pixel com a imagem Z_{i+1} sendo esta relação guardada numa lista de vetores. Sempre que um dado valor x da imagem Z_i é relacionado com um valor y da imagem Z_{i+1} , o valor x é adicionado ao vetor y da lista de vetores. Este processo repete-se para cada pixel das imagens (os pixels pertencentes ao fundo das imagens não são considerados). Seguidamente a lista de vetores é processada, sendo cada vetor ordenado de forma decrescente por número de ocorrências. Deste modo, a lista de vetores relaciona os segmentos das duas imagens da seguinte forma: considerando que um determinado segmento α da imagem Z_{i+1} apresenta um maior nível de intersecção com o segmento β da imagem Z_i , o vetor localizado no índice correspondente ao valor α terá na sua primeira posição o valor β . Após o processamento da lista de vetores, os valores dos segmentos da imagem Z_{i+1} são substituídos pelos valores contidos na lista.

O procedimento repete-se para cada par de imagens consecutivas de Z , sendo que os valores dos segmentos contidos na imagem Z_i são tidos em conta para a imagem Z_{i+1} . Deste modo é criada uma sequência de valores que aumenta de *frame* para *frame* e que permite identificar a origem de cada bactéria presente numa determinada amostra. A ilustração seguinte exemplifica o resultado obtido pelo procedimento descrito.

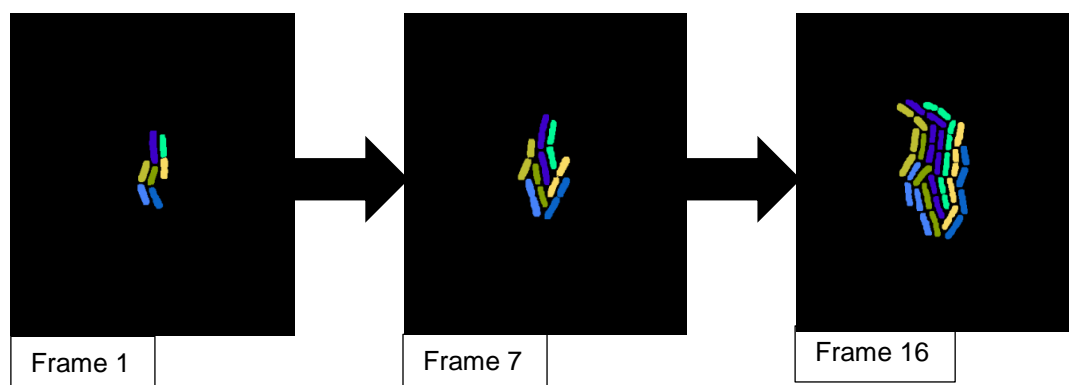


Figura 3.29: Exemplo do resultado da aplicação do processo de seguimento automático.

3.7.3 Correção do processo de seguimento automático

Analisando os resultados obtidos pelo processo de seguimento automático anteriormente descrito verifica-se que, por vezes, existem algumas falhas no seguimento das

bactérias. Uma das principais razões para tal tem a ver com o facto de que, em determinadas sequências de imagens, os segmentos modificam significativamente a sua localização de *frame* para *frame*. Tendo isto em consideração, foi seguida uma estratégia de avaliação da quantidade de segmentos detectados em cada *frame*. Note-se que os segmentos de cada *frame* estão agrupados de acordo com sistema de seguimento. Por exemplo, no *frame* 7, todos os segmentos etiquetados com o valor 5 tiveram como ponto de origem o segmento com valor 5 visualizado no *frame* 1. Deste modo, a partir do processo de *Tracking*, são formados tantos grupos de segmentos quantos os segmentos detectados no primeiro *frame*. O procedimento de correção aqui descrito tem em conta estes mesmos grupos.

Para cada *frame* é processado o número de segmentos contidos em cada grupo. Ao processar a imagem Z_{i+1} , o número de segmentos contabilizado na imagem Z_i é tido em conta. Note-se que quando dois grupos A e B estão geograficamente próximos entre si, o seu número de segmentos não pode diminuir de um *frame* para o *frame* seguinte. Quando um dado segmento x da imagem Z_{i+1} é considerado inicialmente como pertencente ao grupo A (sendo este o grupo para o qual existe uma maior interseção), se for verificado que, em comparação com o *frame* anterior, o grupo B apresenta menos um segmento, verifica-se que teoricamente há uma falha no processo de *Tracking*. Neste caso, o valor do segmento x é reprocessado, sendo verificado se o grupo B é o segundo grupo com maior interseção relativamente ao segmento x . Este processo é suportado pela lista de vetores anteriormente descrita. Se isto for verificado, o valor do segmento é alterado de forma a ser integrado no grupo B. A **Figura 3.30** exemplifica o resultado da aplicação deste procedimento. Este procedimento é sempre realizado para cada par de imagens consecutivas.

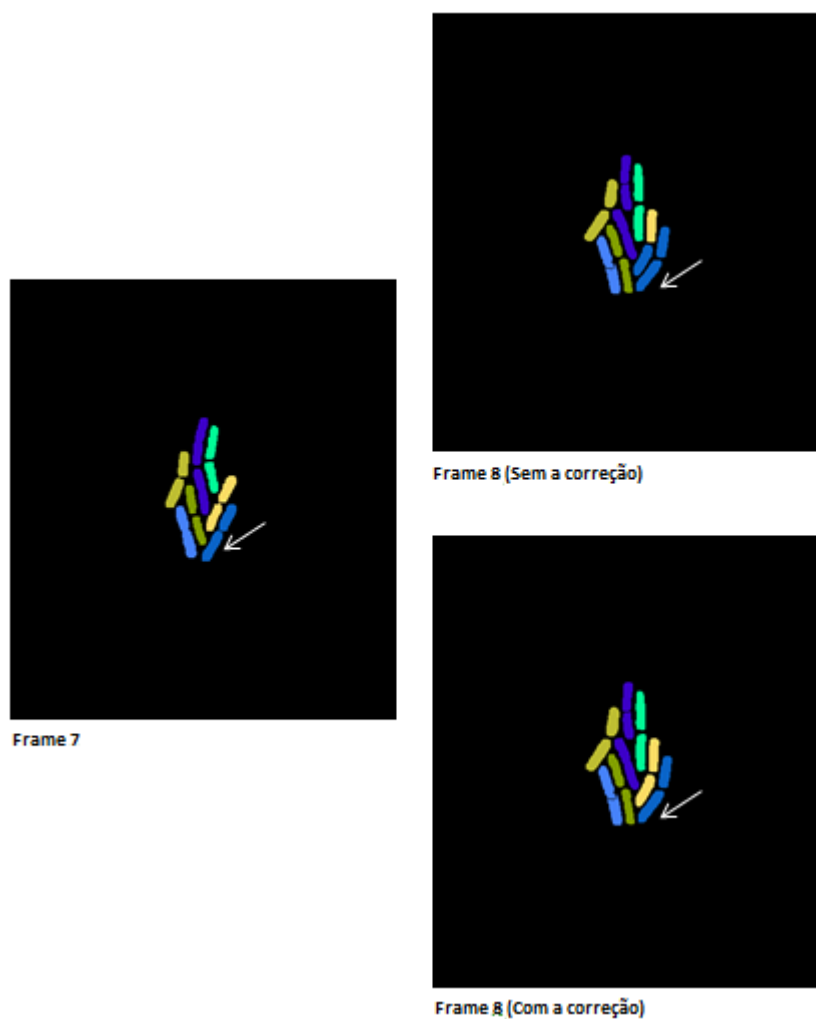


Figura 3.30: Exemplo do resultado do procedimento de correcção do seguimento (sequência de imagens *Schnitzcells*).

4 Resultados obtidos

4.1 Aplicação do modelo de segmentação às imagens disponibilizadas

4.1.1 Procedimento geral

De forma a avaliar a eficiência do algoritmo implementado neste projeto, a aplicação desenvolvida foi testada numa sequência de imagens. As imagens em estudo foram selecionadas a partir das sequências de imagens descritas no **capítulo 3.1**. Foram selecionadas cinco imagens de cada uma das cinco sequências disponibilizadas. Esta seleção não foi, no entanto, realizada ao acaso. Em cada sequência a seleção das cinco imagens foi feita de forma a que toda a sequência temporal fosse parcialmente abrangida, desde a primeira imagem até à última. Algumas destas imagens são seguidamente apresentadas.

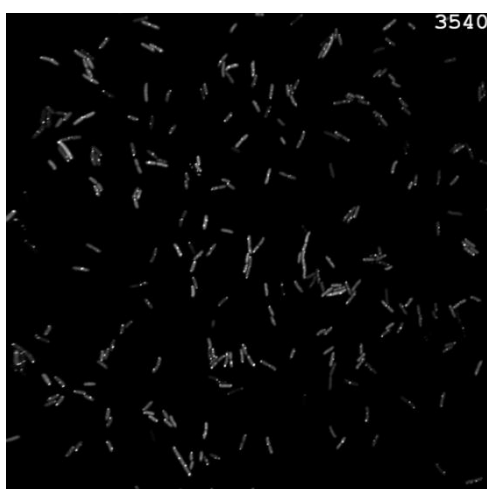


Figura 4.2: Sequência I-Imagem 60.



Figura 4.1: Sequência II-Imagem 38



Figura 4.4: Sequência III-Imagem 45



Figura 4.3: Sequência IV-Imagem 72

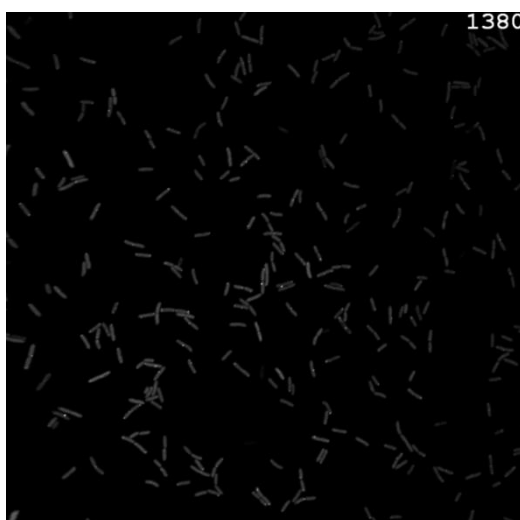


Figura 4.5: Sequência V-Imagem 24

As várias etapas do método desenvolvido neste projeto foram já descritas no **capítulo 3**. Estas etapas são seguidamente resumidas pelo **fluxograma 4.6**. Na ilustração são indicadas algumas especificações relativas a alguns parâmetros de cada procedimento.

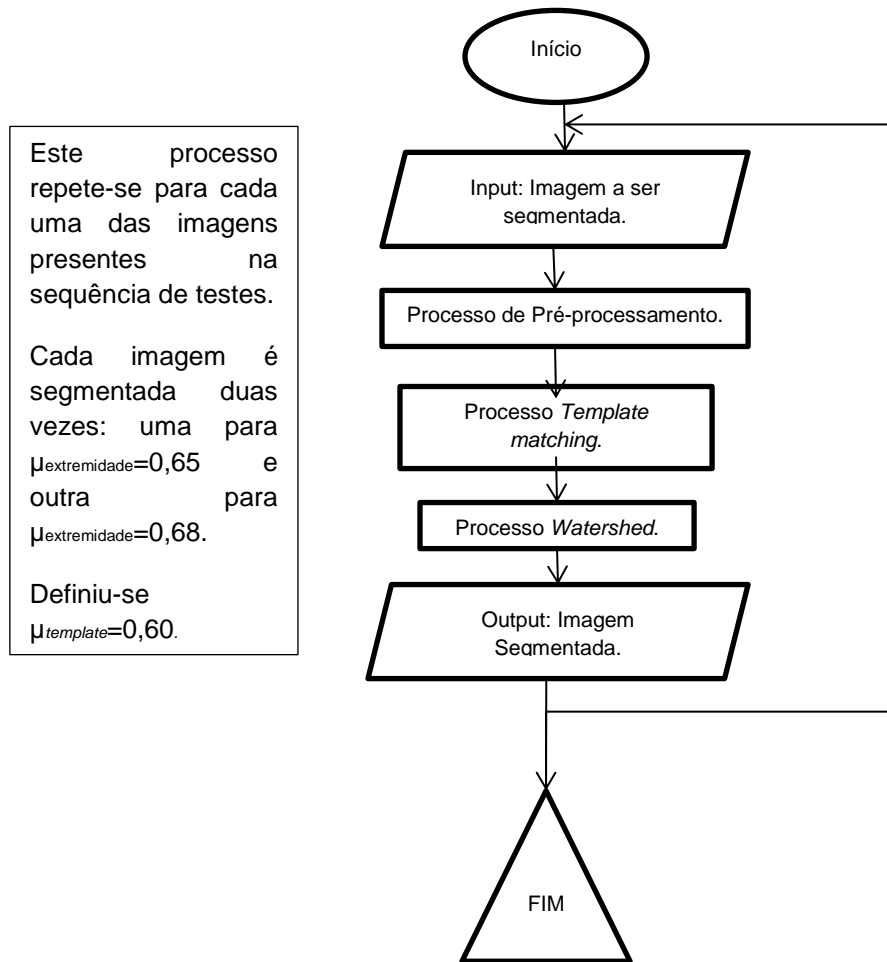


Figura 4.6: Fluxograma relativo aos passos gerais do algoritmo implementado.

Os resultados do processo de segmentação (para $\mu_{extremidade} = 0,68$), das imagens acima ilustradas, são apresentados na seguinte figura.

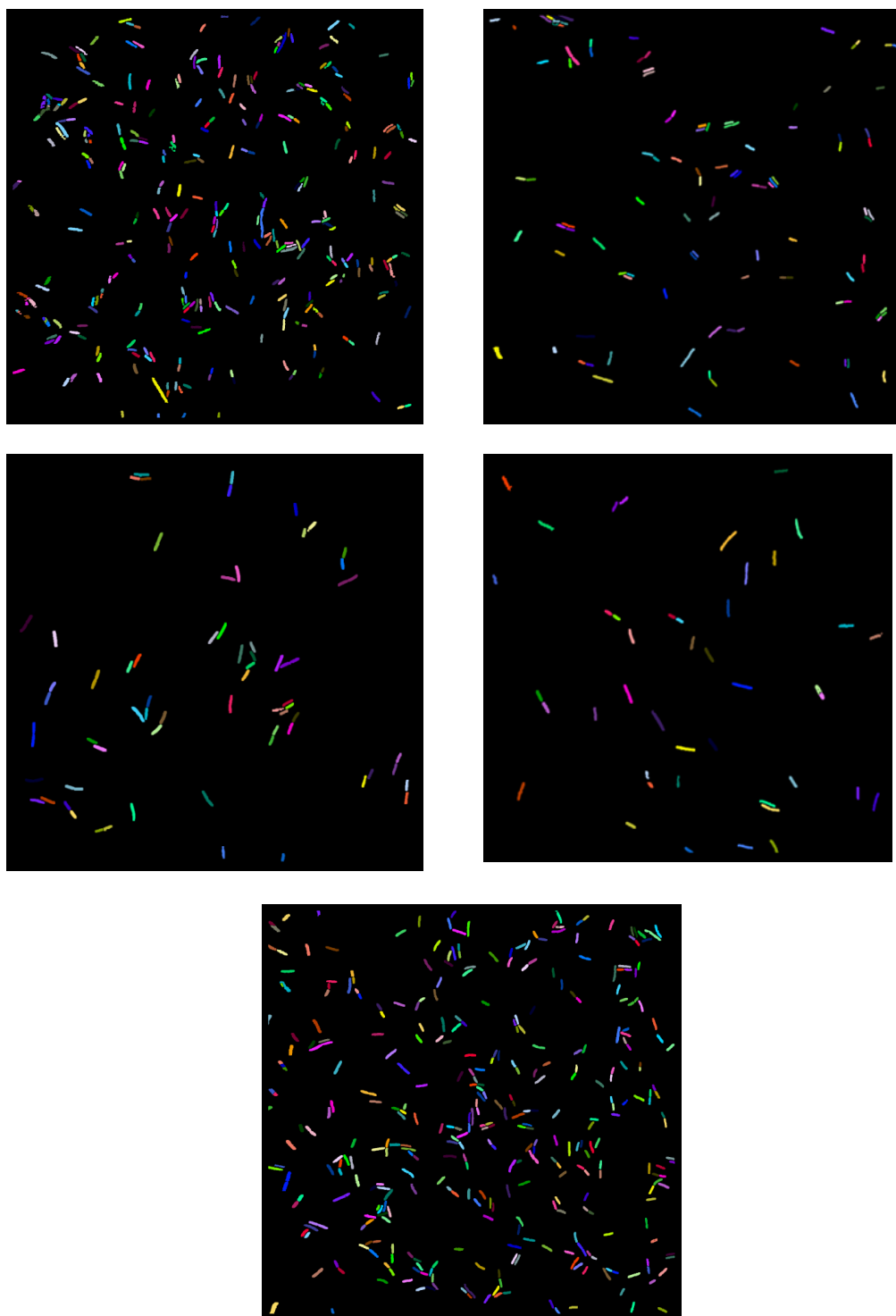


Figura 4.7: Resultados do algoritmo de segmentação implementado quando aplicado às imagens ilustradas nas Figuras 4.1 a 4.5.

4.1.2 Resultados Obtidos

Neste ponto é importante estabelecer uma comparação entre o número de bactérias contabilizado manualmente e o número de bactérias obtido de forma automática, através do procedimento de segmentação (rever o **capítulo 3**). Os resultados obtidos podem ser visualizados na **tabela 4.2**. Tendo em conta o elevado número de bactérias presentes nas imagens em estudo e de forma a que a contagem manual seja feita de forma mais eficiente, é implementado um procedimento gráfico que auxilia a contagem por parte do utilizador. Este procedimento permite ao utilizador colocar pequenas marcas em cada bactéria (com o auxílio do cursor). À medida que estas marcas são colocadas, a contagem é feita automaticamente. Na figura que se segue é possível visualizar um exemplo do processo aqui descrito.

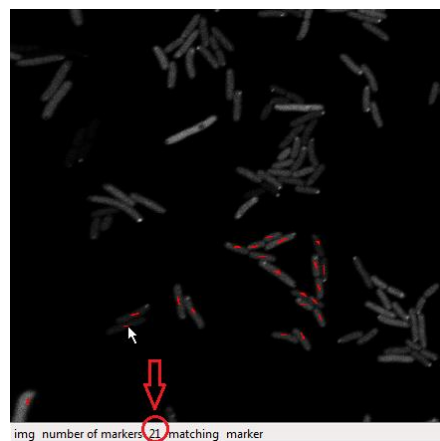


Figura 4.8: Aspeto da interface gráfica criada para suporte ao processo de contagem manual.

A tabela seguinte contém o número de bactérias contabilizado manualmente e automaticamente. A contagem automática apresenta duas variantes: $\mu_{extremidade} = 0,65$ e $\mu_{extremidade} = 0,68$ (**Capítulo 3.3**).

Tabela 4.1: Número de bactérias presentes em cada imagem analisada.

	Imagem	1	2	3	4	5	6	7	8	9	10
Manual		241	247	266	310	444	88	108	116	144	143
Automático	0.65	272	275	281	327	476	112	114	120	158	155
	0.68	256	252	269	316	451	98	107	116	149	150
		Sequência I					Sequência II				
	Imagem	11	12	13	14	15	16	17	18	19	20
Manual		46	45	52	63	72	37	38	43	43	50
Automático	0.65	48	49	55	68	77	39	41	43	51	57
	0.68	46	46	53	63	74	37	39	43	47	52
		Sequência III					Sequência IV				
	Imagem	21	22	23	24	25					
Manual		324	345	361	406	604					
Automático	0.65	353	370	388	436	635					
	0.68	333	352	367	413	608					
		Sequência V									
	Seq. Imagem	Total Seq. I	Total Seq. II	Total Seq. III	Total Seq. IV	Total Seq. V	Total				
Manual		1508	599	278	211	2040	4636				
Automático	0.65	1631	659	297	231	2182	5000				
	0.68	1544	620	282	218	2073	4737				

Na tabela seguinte apresentam-se os valores obtidos através do cálculo do módulo das diferenças (MD) entre o número de bactérias detectado automaticamente (ND) e o número real de bactérias (NR).

$$MD = |ND - NR|$$

Tabela 4.2: Módulo das diferenças entre o número de bactérias reais e o número detectado automaticamente.

	Imagem	1	2	3	4	5	6	7	8	9	10
MD	0.65	31	28	15	17	32	24	6	4	14	12
	0.68	15	5	3	6	7	10	1	0	5	7
		Sequência I					Sequência II				
	Imagem	11	12	13	14	15	16	17	18	19	20
MD	0.65	2	4	3	5	5	2	3	0	8	7
	0.68	0	1	1	0	2	0	1	0	4	2
		Sequência III					Sequência IV				
	Imagem	21	22	23	24	25					
MD	0.65	29	25	27	30	31					
	0.68	9	7	6	7	4					
		Sequência V									

Na tabela seguinte apresentam-se os cálculos relativos às taxas de detecção (TD) de cada imagem em estudo. Esta taxa é calculada através da razão entre o número detectado automaticamente (ND) e o número real de bactérias (NR).

$$TD = \frac{ND}{NR}$$

Tabela 4.3: Taxas de detecção (TD) entre o número real de bactérias (NR) e o número detectado automaticamente (ND).

	Imagem	1	2	3	4	5	6	7	8	9	10
TD	0,65	0,89	0,90	0,95	0,95	0,93	0,79	0,95	0,97	0,91	0,92
	0,68	0,94	0,98	0,99	0,98	0,98	0,90	0,99	1,00	0,97	0,95
Sequência I						Sequência II					

	Imagem	11	12	13	14	15	16	17	18	19	20
TD	0,65	0,96	0,92	0,95	0,93	0,94	0,95	0,93	1,00	0,84	0,88
	0,68	1,00	0,98	0,98	1,00	0,97	1,00	0,97	1,00	0,91	0,96
Sequência III						Sequência IV					

	Imagem	21	22	23	24	25
TD	0,65	0,92	0,93	0,93	0,93	0,95
	0,68	0,97	0,98	0,98	0,98	0,99
		Sequência V				

	Seq.	I	II	III	IV	V	TOTAL
Valor médio(%)	0,65	92,2±1,3	90,7±3,2	93,7±0,7	91,9±2,7	93,3±0,5	92,4±0,8
	0,68	97,5±0,9	96,2±1,8	98,7±0,6	97±1,6	98,3±0,3	97,5±0,5

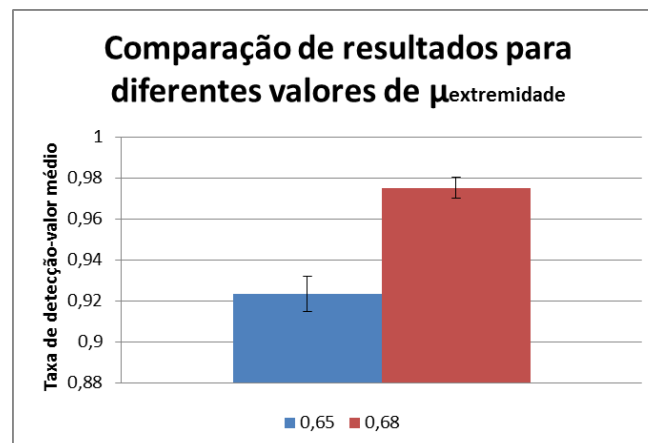


Figura 4.9: Comparação da taxa de detecção média para o conjunto das 25 imagens analisadas.

A tabela seguinte permite avaliar alguns critérios de eficiência do modelo de segmentação. Estes critérios básicos permitem uma melhor percepção relativa à validação do processo de identificação dos diferentes segmentos. Esta avaliação é feita para $\mu_{extremidade} = 0,68$.

Tabela 4.4: Avaliação de critérios de eficiência do modelo de segmentação.

Imagem	1	2	3	4	5	6	7	8	9	10
Caso A	11	5	9	4	15	13	0	0	6	7
Caso B	4	1	1	1	8	3	1	0	1	0
Sequência I					Sequência II					
Imagem	11	12	13	14	15	16	17	18	19	20
Caso A	0	1	1	0	2	0	1	1	4	1
Caso B	0	0	0	0	0	0	0	1	0	0
Sequência III					Sequência IV					
Imagem	21	22	23	24	25					
Caso A	9	9	7	7	8					
Caso B	0	1	1	0	4					
Sequência V										

Caso A - Objectos sobressegmentados (objetos que são "partidos" quando na realidade não o deveriam ser).

Caso B - Objectos subsegmentados (objetos que não são "partidos" quando na realidade deveriam ser).

4.1.3 Análise de resultados

O algoritmo de segmentação desenvolvido apresenta uma taxa de detecção (TD) média de $92,352\% \pm 0,844\%$ para $\mu_{extremidade} = 0,65$ e de $97,523\% \pm 0,511\%$ para $\mu_{extremidade} = 0,68$, tendo em conta o conjunto de imagens testado. É possível verificar esta situação através da representação gráfica apresentada na **Figura 4.9**. As taxas de detecção para cada sequência são apresentadas na **Tabela 4.3**. Os resultados obtidos podem considerar-se satisfatórios, principalmente para $\mu_{extremidade} = 0,68$, o que pode ser constatado pela **Tabela 4.2**, que ilustra o módulo das diferenças relativo ao número de bactérias.

A **Tabela 4.4** apresenta dados relativos a um critério específico de avaliação da eficiência da segmentação realizada, para $\mu_{extremidade} = 0,68$. O caso A representa as situações em que, por exemplo, uma bactéria é segmentada de forma a formar duas bactérias, quando na verdade deveria apenas formar uma. O caso B representa os casos em que, por exemplo, duas bactérias são segmentadas como se formassem um único segmento, quando na realidade são dois. Analisando os dados aqui fornecidos, entende-se que ocorrem mais casos do tipo A do que casos do tipo B, sendo os de tipo B bastante raros. Na sequência I e na sequência V são evidenciados mais casos de incorreção. Este facto prende-se essencialmente com o elevado número de colónias de bactérias que estas imagens possuem, sendo mais provável existirem casos de incorreção. Ainda assim, os resultados apresentados são bastante satisfatórios.

Repare-se que, para a sequência II ($96,170\% \pm 1,798\%$), a taxa de detecção (TD) apresenta um valor menos significativo. A diferença entre o número de bactérias detectado manualmente e o número de bactérias detectado automaticamente, prende-se essencialmente com dois pontos principais:

1. Elevada variação em termos de tamanho e forma geométrica das bactérias visualizadas nas imagens;
2. Baixa iluminação e elevado grau de desfocagem das primeiras imagens de cada sequência.

Estes pontos são principalmente notados na sequência II, nomeadamente na imagem 6, o que contribui para o valor apresentado. A **Tabela 4.2**, relativa ao módulo das diferenças, também evidencia esta situação.

As variações nas características das imagens que se verificam de sequência para sequência influenciam negativamente os resultados obtidos. No entanto, apesar deste facto, o algoritmo desenvolvido apresenta resultados bastante positivos, como pode ser comprovado pelos dados estatísticos aqui fornecidos. O próximo passo prende-se com a necessidade de avaliar a versatilidade do algoritmo desenvolvido, estabelecendo uma comparação com os resultados obtidos por um dos softwares com maiores índices de eficiência já existentes.

4.2 Imagens *Schnitzcells*

4.2.1 Procedimento geral

Neste capítulo será analisada a aplicação do algoritmo de segmentação em imagens provenientes de outra fonte. Este teste serve para complementar os testes realizados nas imagens *standard* (analisadas no **capítulo 4.1**) e ao mesmo tempo, avaliar a adaptação do software criado neste projeto a imagens que apresentam algumas variações em termos das suas características básicas. Por outro lado, esta análise permite a comparação dos resultados obtidos, com os obtidos por um software existente no mercado. O software *Schnitzcells* é considerado um dos mais eficientes softwares de segmentação de imagem (ver **capítulo 1.5**). As imagens aqui em estudo e os respectivos resultados através do software *Schnitzcells* são publicamente disponibilizados (<http://cell.caltech.edu/Schnitzcells/>).

Nesta comparação aplicou-se, em geral, a mesma metodologia que foi aplicada na sequência de imagens anterior (ver **fluxograma 4.6**). As ilustrações que se seguem são referentes ao resultado do processo de segmentação através do software *Schnitzcells* aplicado à sequência de imagens em estudo.

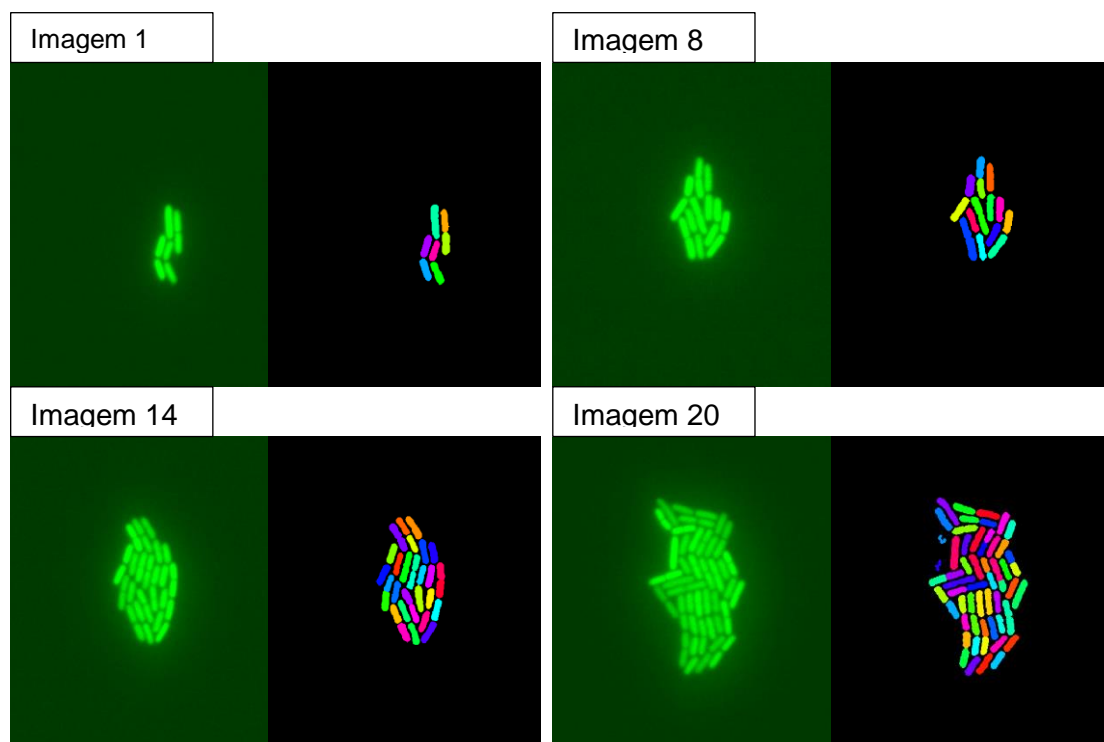


Figura 4.10: Algumas das imagens “*Schnitzcells*” e respetivo resultado de segmentação através do software “*Schnitzcells*”.

São seguidamente apresentados os resultados obtidos com o software desenvolvido neste projeto ($\mu_{extremidade} = 0,68$).

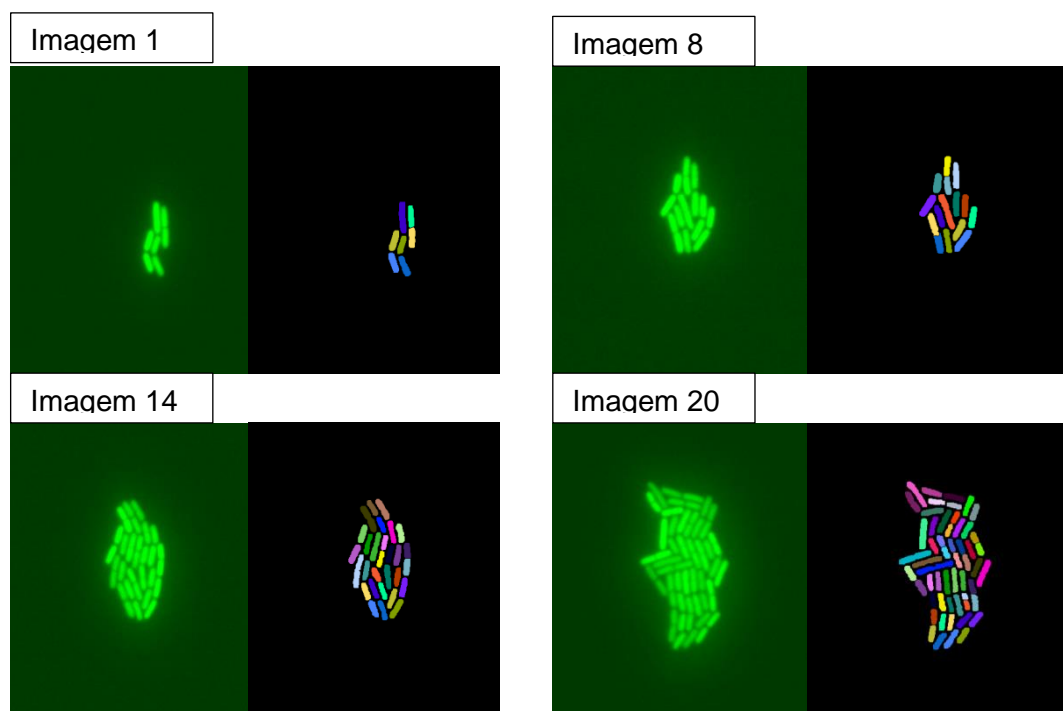


Figura 4.11: Algumas imagens resultantes da segmentação através do software desenvolvido neste projeto.

4.2.2 Resultados Obtidos

A tabela seguinte permite estabelecer uma comparação entre os resultados obtidos em ambos os softwares, sendo contabilizado o número de bactérias detectado por ambos os softwares.

Tabela 4.5: Número de bactérias presente em cada uma das 20 imagens “Schnitzcells”

Imagem	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Manual	7	7	7	8	10	13	14	15	16	17	20	27	28	30	34	36	39	43	47	54
Schnitzcells	7	7	7	8	9	11	14	14	17	17	19	23	26	28	34	35	39	41	48	55
Automático(0,68)	7	7	7	8	10	13	14	15	16	17	21	27	28	30	34	36	39	44	48	55

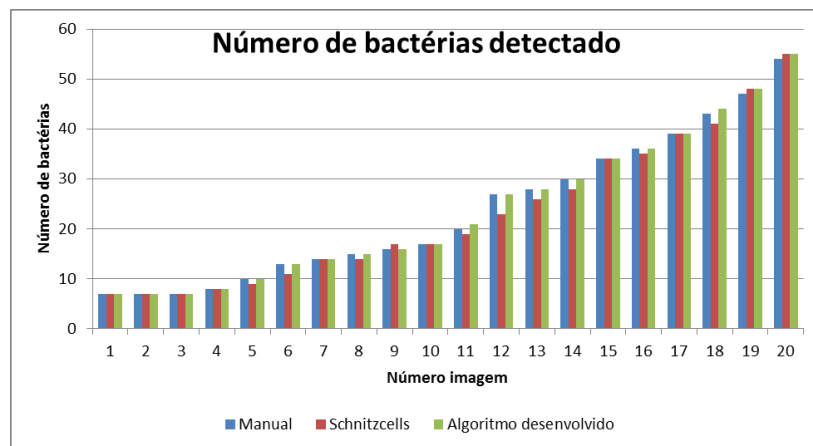


Figura 4.12: Gráfico ilustrativo do número de bactérias detectado.

Na tabela seguinte apresentam-se os valores obtidos através do cálculo do módulo das diferenças (MD).

$$MD = |ND - NR|$$

Tabela 4.6: Módulo da diferença (MD) entre o número de bactérias real (NR) e o número de bactérias detectado (ND), para cada algoritmo.

Imagem	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MD Schnitzcells	0	0	0	0	1	2	0	1	1	0	1	4	2	2	0	1	0	2	1	1
MD Auto(0,68)	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1

A seguinte tabela permite avaliar determinados critérios de eficiência do modelo de segmentação, tal como foi realizado na secção anterior.

Tabela 4.7: Avaliação de critérios de eficiência do modelo de segmentação.

Imagem	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Schnitzcells	Caso A	0	0	0	0	0	0	0	1	1	2	1	0	0	0	1	0	0	1	2
	Caso B	0	0	0	0	1	2	0	1	0	1	5	2	0	0	1	0	3	0	1
Auto(0,68)	Caso A	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	2
	Caso B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2

Caso A - Objectos sobressegmentados (objectos que são "partidos" quando na realidade não o deveriam ser).

Caso B - Objectos subssegmentados (objectos que não são "partidos" quando na realidade deveriam ser).

Na tabela seguinte apresentam-se os cálculos relativos às taxas de detecção (TD) de cada imagem em estudo.

$$TD = \frac{ND}{NR}$$

Tabela 4.8: Taxas de detecção (TD) entre o número real de bactérias (NR) e o número detectado automaticamente (ND).

TD	Imagem	1	2	3	4	5	6	7	8	9	10
	Schnitzcells	1,00	1,00	1,00	1,00	0,90	0,85	1,00	0,93	0,94	1,00
	Auto(0,68)	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00

TD	Imagem	11	12	13	14	15	16	17	18	19	20
	Schnitzcells	0,95	0,85	0,93	0,93	1,00	0,97	1,00	0,95	0,98	0,98
	Auto(0,68)	0,95	1,00	1,00	1,00	1,00	1,00	1,00	0,98	0,98	0,98

Valor médio (%)	Schnitzcells	95,9±1,1
	Auto(0,68)	99,5±0,3

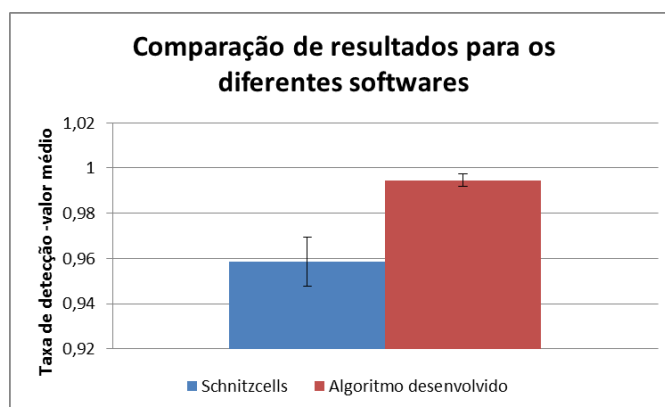


Figura 4.13: Comparação da taxa de detecção média para o conjunto das 20 imagens analisadas.

4.2.3 Análise de resultados

Através da análise dos dados estatísticos visualizados na secção anterior, é possível tirar algumas ilações importantes (ver **Tabela 4.5** e **Figura 4.13**). Começamos por olhar a taxa de detecção (TD) apresentada em cada uma das vinte imagens disponibilizadas (ver **Tabela 4.8**). Note-se que a TD do algoritmo aqui desenvolvido é praticamente sempre superior à registada no software *Schnitzcells*, para cada uma das vinte imagens. Algumas incorreções visualizadas (ver **Tabela 4.6**) para estas imagens prendem-se essencialmente com os elevados níveis de ruído apresentados, aliada a uma relativa diminuição de contraste em determinadas regiões das imagens. O algoritmo desenvolvido apresenta uma taxa de detecção média de 99,453%±0,279% (ver **Figura 4.13**), um resultado bastante satisfatório, principalmente quando comparado com os 95,856%±1,087% obtidos para o software *Schnitzcells*.

A **Tabela 4.4** apresenta dados relativos a um critério específico de avaliação da eficiência da segmentação realizada. Tal como se pode constatar, na maioria das imagens, são apresentados mais casos de incorreção, tanto para o caso *A* como para o caso *B*, para o software *Schnitzcells* do que para o algoritmo aqui descrito.

4.3 Seguimento automático

O algoritmo de seguimento automático foi aplicado a três sequências de imagens: as imagens *Schnitzcells* (ver **capítulo 4.2**) e duas das sequências restantes que apresentam-se como duas das sequências introduzidas no **capítulo 4.1** (Sequência I e V). As sequências I e V foram escolhidas por conterem um maior número de bactérias (apresentando uma maior complexidade) e por serem as sequências onde é possível visualizar uma maior variação espacial entre *frames*.

As imagens *Schnitzcells* (**Figuras 4.10 e 4.11**), são caracterizadas por uma elevada variação de *frame* para *frame*, no que diz respeito à localização de cada segmento. Este filme apresenta 19 *frames*, sendo que a tabela seguinte contém o número de incorreções visualizadas entre *frames* consecutivos.

Tabela 4.9: Número de incorreções detectado entre *frames* consecutivos para o seguimento da sequência *Schnitzcells*.

Imagem	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
nº incorreções	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	2

Quando comparadas com a sequência *Schnitzcells*, as restantes sequências em estudo apresentam características distintas. A variação de segmentos é muito reduzida de *frame* para *frame*, sendo que a resolução temporal dos filmes apresentados é relativamente elevada. Cada sequência em estudo apresenta 82 *frames*. Através da análise dos resultados obtidos, o número de incorreções observado é apresentado na tabela seguinte.

Tabela 4.10: Número de incorreções ao longo dos dois filmes e percentagem de erro.

Filme	I	II
nº incorreções	11	6
Percentagem de erro	4,911%	1,786%

As percentagens de erro são definidas através da razão entre o número de incorreções encontrado ao longo dos 82 *frames* e número de segmentos detectados no primeiro *frame*. Estes valores são bastante razoáveis, tendo em conta que existe já um erro associado ao processo prévio de segmentação de cada uma das imagens. As incorreções verificadas são, na sua maioria, uma consequência de situações de sobressegmentação entre *frames* consecutivos, o que influencia claramente o posterior processo de seguimento. Tal como se pode verificar através dos resultados obtidos, o algoritmo de seguimento apresenta um grau de eficiência bastante satisfatório, podendo ser futuramente desenvolvido e aplicado a várias sequências de imagens celulares.

5 Conclusão

5.1 Considerações finais

O projeto desenvolvido consiste na criação de um modelo geral de segmentação para aplicação a imagens microscópicas. As imagens em estudo foram disponibilizadas pelo laboratório *Laboratory of Biosystem Dynamics da Tampere University of Technology*, contendo várias dezenas de colónias do tipo *E.coli*. As características dessas imagens podem ser consultadas no **capítulo 3.1**. O modelo desenvolvido é constituído por uma técnica de segmentação baseada no método *Watershed*, através de um sistema de marcas que localizam cada um dos segmentos a serem identificados. Após a aplicação de um pré-processamento específico sobre as imagens em estudo, as marcas são adquiridas através de um sistema de *Template matching* invariante relativamente a várias propriedades dos objetos, como escalamento, rotação e variação de intensidade.

As técnicas de pré-processamento introduzidas têm como objetivo fundamental simplificar as propriedades das imagens em consideração, ou seja, alterar parte da sua informação, de forma a facilitar a extração de características que tornam o posterior processo de segmentação mais eficiente. A sequência de transformações aplicada permite solucionar parcialmente os problemas levantados por algumas das características intrínsecas das imagens. Destacam-se as grandes variações de intensidade visualizadas em diferentes regiões de cada uma das imagens, o reduzido contraste em zonas com maior presença de contornos e a elevada complexidade das várias colónias presentes nas imagens.

No processo de *Template matching* implementado são utilizados cinco templates previamente definidos, tendo em conta as variações das dimensões dos objetos a serem identificados. Para cada template é executado um procedimento de rotação que permite um sistema robusto de detecção, independente das orientações espaciais dos vários segmentos. Adicionalmente, é aplicado um template alternativo que identifica as extremidades dos vários segmentos, o que auxilia na determinação de marcas individuais para objetos demasiadamente próximos entre si. Este processo assume grande importância tendo em conta a já enunciada complexidade, evidenciada pela densidade das diversas colónias existentes.

Com base nas marcas identificativas de cada objeto, é iniciado um procedimento de transformação destas marcas, que é finalizado pela aplicação de uma técnica de *Watershed*, responsável pelo processo de segmentação e consequente contagem automática das bactérias presentes nas imagens em estudo. A aplicação do algoritmo desenvolvido nas imagens disponibilizadas obtém uma percentagem de êxito de $97,523\% \pm 0,511\%$, o que se pode

considerar um resultado bastante satisfatório, tendo em conta as propriedades das imagens em questão. A eficiência deste modelo é evidenciada através da comparação dos seus resultados com os obtidos por um software de segmentação já existente sobre um conjunto distinto de imagens celulares. Os resultados obtidos através deste processo são bastante positivos, obtendo-se uma percentagem de $99,453\% \pm 0,279\%$, quando aplicado à sequência descrita na secção 4.2.

Por fim, relativamente ao desenvolvimento do algoritmo de seguimento automático, a observação das várias sequências de imagens disponibilizadas e a análise dos resultados do processo de segmentação anteriormente descrito, permitem a criação de uma base sólida e simples, relativa à melhor forma de desenvolver um algoritmo que permita uma fácil observação das características de reprodução e mobilidade celular. O algoritmo desenvolvido é processado em três passos: inicialmente é feita uma correção da segmentação já realizada, através de um procedimento de comparação entre frames consecutivos. Seguidamente é executado um processo de seguimento baseado num procedimento de interseção de frames consecutivos, sendo, por último, realizado um procedimento de correção do processo anterior. Os resultados obtidos apresentam um grau de eficiência bastante satisfatório.

Ao longo deste projeto foram encontrados alguns pontos que podem ser melhorados, tendo em conta alguns desenvolvimentos futuros que são seguidamente apresentados.

5.2 Desenvolvimentos futuros

O algoritmo implementado apresenta pequenas deficiências que podem ser colmatadas num desenvolvimento futuro. O tempo de execução do algoritmo é um dos principais pontos de interesse. A técnica de *Template matching* implementada assume-se como sendo significativamente exigente a nível computacional, consumindo por vezes um elevado tempo de processamento. O melhoramento do tempo de processamento torna-se-ia altamente vantajoso, permitindo até uma mais simples implementação de *templates* com um maior número de variações, nomeadamente, a implementação de um processo de rotação com um nível de resolução mais elevado.

As imagens disponibilizadas apresentam elevadas variações no que diz respeito à forma geométrica das bactérias a serem detectadas. Ainda assim, o modelo desenvolvido apresenta resultados bastante bons no que diz respeito a problemas de sobressegmentação, tal como pode ser constatado pelo conjunto de testes efetuado. Contudo, este processo pode ainda assim ser melhorado, através da aplicação posterior de uma técnica de classificação e correção de segmentos. Esta técnica pode ter em conta as características morfológicas dos próprios segmentos.

Após o processo de localização e posterior contagem automática de bactérias, é possível executar diversos estudos relativos à morfologia e ao comportamento celular. Neste sentido, é possível desenvolver técnicas de avaliação automática que permitem, de forma eficiente, criar ferramentas essenciais na investigação de diversos processos biológicos.

6 Bibliografia

- [1] G. A. Baxes, *Digital image processing: principles and applications*. Wiley, 1994.
- [2] Q. Wu, F. A. Merchant, and K. R. Castleman, *Microscope Image Processing*. Academic Press, 2008.
- [3] G. J. Tortora, B. R. Funke, and C. L. Case, *Microbiology: An Introduction: Books a La Carte Edition*. Prentice Hall, 2012.
- [4] C. Solomon and T. Breckon, *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*. John Wiley & Sons, 2011.
- [5] A. Mosig, S. Jäger, C. Wang, S. Nath, I. Ersoy, K. Palaniappan, and S.-S. Chen, “Tracking cells in Life Cell Imaging videos using topological alignments,” *Algorithms for molecular biology : AMB*, vol. 4, p. 10, Jan. 2009.
- [6] N. Ray, S. T. Acton, and K. Ley, “Tracking leukocytes in vivo with shape and size constrained active contours,” *IEEE transactions on medical imaging*, vol. 21, no. 10, pp. 1222–35, Oct. 2002.
- [7] Q. Wang, J. Niemi, C. Tan, L. You, and M. West, “Image Segmentation and Dynamic Lineage Analysis in Single-Cell Fluorescence Microscopy,” *Image Processing*, 2010.
- [8] M. Analysis, A. Sacan, H. Ferhatosmanoglu, and H. Coskun, “CellTrack : An Open-Source Software for Cell Tracking,” vol. 3, pp. 2–4, 2008.
- [9] Q. Wang, L. You, and M. West, “CellTracer : Software for automated image segmentation and lineage mapping for single-cell studies,” pp. 1–2, 2008.
- [10] C. Du and T. Bretschneider, “Updated CellTracker software manual software is based Table of Contents,” no. 2006, pp. 787–794, 2010.
- [11] J. Klein, S. Leupold, I. Biegler, R. Biedendieck, R. Münch, and D. Jahn, “TLM-Tracker: Software for cell segmentation, tracking and lineage analysis in time-lapse microscopy movies,” *Bioinformatics (Oxford, England)*, 2012.

- [12] J. W. Young, J. C. W. Locke, A. Altinok, N. Rosenfeld, T. Bacarian, P. S. Swain, E. Mjolsness, and M. B. Elowitz, "Measuring single-cell gene expression dynamics in bacteria using fluorescence time-lapse microscopy.," *Nature protocols*, vol. 7, no. 1, pp. 80–8, Jan. 2012.
- [13] P. Vallotton, L. Mililli, L. Turnbull, and C. Whitchurch, "Segmentation of Dense 2D Bacilli Populations," *2010 International Conference on Digital Image Computing: Techniques and Applications*, pp. 82–86, Dec. 2010.
- [14] S. Osher, R. Fedkiw, and K. Piechor, *Level Set Methods and Dynamic Implicit Surfaces*, vol. 57, no. 3. 2004, p. B15.
- [15] R. Tsai and S. Osher, "Level set methods and their applications in image science," pp. 1–37, 2003.
- [16] K.S. Fu and J.K. Mui, "A survey on image segmentation," *Pattern Recognition*, pp. 3 – 16, 1981.
- [17] H. Daims, S. Lückner, and M. Wagner, "Daime, a Novel Image Analysis Program for Microbial Ecology and Biofilm Research.," *Environmental microbiology*, vol. 8, no. 2, pp. 200–13, Feb. 2006.
- [18] O. J. Tobias and R. Seara, "Image segmentation by histogram thresholding using fuzzy sets.," *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 11, no. 12, pp. 1457–65, Jan. 2002.
- [19] J. Slavík, *Fluorescence Microscopy and Fluorescent Probes*, no. vol. 1. Springer, 1996.
- [20] M. FORERO, F. SROUBEK, and G. CRISTOBAL, "Identification of tuberculosis bacteria based on shape and color," *Real-Time Imaging*, vol. 10, no. 4, pp. 251–262, Aug. 2004.
- [21] R. a Russell, N. M. Adams, D. a Stephens, E. Batty, K. Jensen, and P. S. Freemont, "Segmentation of fluorescence microscopy images for quantitative analysis of cell nuclear architecture.," *Biophysical journal*, vol. 96, no. 8, pp. 3379–89, Apr. 2009.
- [22] C. L. Viles and M. E. Sieracki, "Measurement of marine picoplankton cell size by using a cooled, charge-coupled device camera with image-analyzed fluorescence microscopy.," *Applied and environmental microbiology*, vol. 58, no. 2, pp. 584–92, Feb. 1992.
- [23] J. Canny, "A computational approach to edge detection.," *IEEE transactions on pattern analysis and machine intelligence*, vol. 8, no. 6, pp. 679–98, Jun. 1986.
- [24] U. De Savoie, "APPROACH OF IMAGE SEGMENTATION," pp. 470–474, 1994.
- [25] Y. Ji, K. H. Chang, and C.-C. Hung, "Efficient edge detection and object segmentation using Gabor filters," *Proceedings of the 42nd annual Southeast regional conference on - ACM-SE 42*, p. 454, 2004.

- [26] A. Shopov, S. C. Williams, and P. G. Verity, "Improvements in image analysis and fluorescence microscopy to discriminate and enumerate bacteria and viruses in aquatic samples," *Aquatic Microbial Ecology*, vol. 22, no. 1991, pp. 103–110, 2000.
- [27] P. Vallotton, C. Sun, D. Wang, L. Turnbull, and C. Whitchurch, "Segmentation and tracking individual *Pseudomonas aeruginosa* bacteria in dense populations of motile cells," *Image (Rochester, N.Y.)*, no. Ivcnz, pp. 221–225, 2009.
- [28] H. Ateş, "An Image-Processing Based Automated Bacteria Colony Counter," pp. 18–23, 2009.
- [29] J. M. Bewes, N. Suchowerska, and D. R. McKenzie, "Automated cell colony counting and analysis using the circular Hough image transform algorithm (CHiTA).," *Physics in medicine and biology*, vol. 53, no. 21, pp. 5991–6008, Nov. 2008.
- [30] Q. Yao, Y. Zhou, and J. Wang, "An automatic segmentation algorithm for touching rice grains images," *2010 International Conference on Audio, Language and Image Processing*, pp. 802–805, Nov. 2010.
- [31] K. Veropoulos, C. Campbell, G. Learmonth, B. Knight, and J. Simpson, "The Automated Identification of Tubercle Bacilli using Image Processing and Neural Computing Techniques," vol. 2, no. September, pp. 797–802, 1998.
- [32] A. Nakib, H. Oulhadj, and P. Siarry, "Microscopic image segmentation with two-dimensional exponential entropy based on hybrid microcanonical annealing," 2007.
- [33] G. Bradski and A. Kaehler, *Learning OpenCV*. 2008.
- [34] E. R. Dougherty and R. A. de Lotufo, *Hands-On Morphological Image Processing*. Spie Press, 2003.
- [35] S. Chen and H. Leung, "Chaotic spread spectrum watermarking for remote sensing images," *Journal of Electronic Imaging*, vol. 13, no. 1, p. 220, 2004.
- [36] K. Fredriksson, *Rotation Invariant Template Matching*. 2001.
- [37] R. Brunelli, *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley, 2009.
- [38] M. Choi and W. Kim, "A novel two stage template matching method for rotation and illumination invariance," vol. 35, pp. 119–129, 2002.
- [39] S. Omachi and M. Omachi, "Fast template matching with polynomials.," *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 16, no. 8, pp. 2139–49, Aug. 2007.
- [40] D. Marimon and T. Ebrahimi, "Efficient rotation-discriminative template matching," pp. 1–11, 2007.

- [41] P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," vol. 15, no. April 1971, pp. 11–15, 1972.
- [42] L. a. F. Fernandes and M. M. Oliveira, "A general framework for subspace detection in unordered multidimensional data," *Pattern Recognition*, vol. 45, no. 9, pp. 3566–3579, Sep. 2012.
- [43] F. O. Gorman, "Finding picture edges through collinearity of feature points," 1976.
- [44] J. Ji, G. Chen, and L. Sun, "A novel Hough transform method for line detection by enhancing accumulator array," *Pattern Recognition Letters*, vol. 32, no. 11, pp. 1503–1510, Aug. 2011.
- [45] Y. Xie, "A new efficient ellipse detection method," vol. 00, no. c, pp. 0–3, 2002.
- [46] D. S. Huang, K. H. Jo, H. H. Lee, H. J. Kang, and V. Bevilacqua, *Emerging Intelligent Computing Technology and Applications. With Aspects of Artificial Intelligence: 5th International Conference on Intelligent Computing, Icic 2009 Ulsan, South Korea, September 16-19, 2009 Proceedings*. Springer, 2009.
- [47] J. P. Lewis, "Fast Normalized Cross-Correlation," vol. 1995, no. 1, 1995.
- [48] C. H. Chen, *Handbook of Pattern Recognition and Computer Vision: 4th Edition*. Imperial College Press, 2010.
- [49] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
- [50] a. C. Berg and J. Malik, "Geometric blur for template matching," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, pp. I–607–I–614.
- [51] F. Ullah and S. Kaneko, "Using orientation codes for rotation-invariant template matching," *Pattern Recognition*, vol. 37, no. 2, pp. 201–209, Feb. 2004.
- [52] H. Schweitzer, J. W. Bell, and F. Wu, "Very Fast Template Matching," in *Computer Vision — ECCV 2002*, vol. 2353, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds. Springer Berlin Heidelberg, 2006, pp. 358–372.
- [53] A. E. Abdel-Hakim and A. a. Farag, "CSIFT: A SIFT Descriptor with Color Invariant Characteristics," *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*, vol. 2, pp. 1978–1983, 2006.
- [54] D. G. Lowe, "Object recognition from local scale-invariant features," *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pp. 1150–1157 vol.2, 1999.
- [55] H. Bay and A. Ess, "Speeded-Up Robust Features (SURF)," vol. 110, pp. 346–359, 2008.

-
- [56] S. Alves, D. Araújo, and H. Yong, "Ciratefi : An RST-Invariant Template Matching with Extension to Color Images," pp. 1–16, 2011.
- [57] S. A. De Araújo and H. Y. Kim, "Color-Ciratefi : A color-based RST-invariant template matching algorithm," 2010.
- [58] U. Nacional and A. Valery, "Automated Characterization of Bacteria in Confocal Microscope Images," 2008.
- [59] C. Teutsch, D. Berndt, E. Trostmann, and M. Weber, "Real-time detection of elliptic shapes for automated object recognition and object tracking," *In J of Cognitive Neuroscience v3 nr1*, vol. 49, no. c, 2006.
- [60] K. Souza, A. Pinheiro, W. Amorim, and V. Odakura, "Contagem automática de unidades formadoras de colónias de bactérias em placas de Petri com o uso do algoritmo Template Matching," 2012.
- [61] P. Soille, "Watersheds in Digital Spaces : An Efficient Algorithm Based on Immersion Simulations," 1991.
- [62] C. Vachier and C. D. M. Math, "The Viscous Watershed Transform," pp. 251–267, 2005.
- [63] S. Beucher, "The watershed transformation applied to image segmentation," pp. 1–26, 1991.
- [64] H. Digabel and C. Lantuéjoul, "Iterative algorithms," *Proc. 2nd European Symposium on Quantitative Analysis of Microstructures in Material Science, Biology and Medicine*, 1977.
- [65] L. Vincent, "Morphological Grayscale Reconstruction in Image Analysis : Applications and Efficient Algorithms," vol. 2, no. 2, 1993.
- [66] C. Lantuejoul and S. Beucher, "On the use of the geodesic metric in image analysis," *Journal of Microscopy*, vol. 121, no. 1, pp. 39–49, 1981.
- [67] P. Soille, *Morphological Image Analysis: Principles and Applications*. Springer, 2010.
- [68] E. Hodneland, N. V Bukoreshtliev, T. W. Eichler, X.-C. Tai, S. Gurke, A. Lundervold, and H.-H. Gerdes, "A unified framework for automated 3-d segmentation of surface-stained living cells and a comprehensive segmentation evaluation," *IEEE transactions on medical imaging*, vol. 28, no. 5, pp. 720–38, May 2009.
- [69] X.-C. Tai, E. Hodneland, J. Weickert, N. V Bukoreshtliev, A. Lundervold, and H.-H. Gerdes, "Level set methods for watershed image segmentation," in *Proceedings of the 1st international conference on Scale space and variational methods in computer vision*, 2007, pp. 178–190.

- [70] J. Serra, L. Vincent, C. D. M. Math, and E. N. Sup, "An overview of morphological filtering," vol. 11, no. 1, 1992.
- [71] M. Grimaud, "New measure of contrast: the dynamics," pp. 292–305, 1992.
- [72] a I. Dow, S. a Shafer, J. M. Kirkwood, R. a Mascari, and a S. Waggoner, "Automatic multiparameter fluorescence imaging for determining lymphocyte phenotype and activation status in melanoma tissue sections," *Cytometry*, vol. 25, no. 1, pp. 71–81, Sep. 1996.
- [73] P. S. Adiga, "Integrated approach for segmentation of 3-D confocal images of a tissue specimen.," *Microscopy research and technique*, vol. 54, no. 4, pp. 260–70, Aug. 2001.
- [74] U. Adiga, R. Malladi, R. Fernandez-Gonzalez, and C. Ortiz de Solorzano, "High-throughput analysis of multispectral images of breast cancer tissue.," *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 15, no. 8, pp. 2259–68, Aug. 2006.
- [75] G. Lin, U. Adiga, K. Olson, J. F. Guzowski, C. a Barnes, and B. Roysam, "A hybrid 3D watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks," *Cytometry. Part A : the journal of the International Society for Analytical Cytology*, vol. 56, no. 1, pp. 23–36, Nov. 2003.
- [76] K. A. N. Jiang, Q. G. Liao, and S. Dai, "A novel white blood cell segmentation scheme using scale-space filtering and watershed clustering," *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693)*, pp. 2820–2825, 2003.
- [77] J. Selinummi, J. Seppala, O. Yli-Harja, and J. A. Puhakka, "Software for quantification of labeled bacteria from digital microscope images by automated image analysis," *BioTechniques*, vol. 39, no. 6, p. 859, 2005.
- [78] X. Chen, X. Zhou, and S. T. C. Wong, "Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy," *IEEE transactions on bio-medical engineering*, vol. 53, no. 4, pp. 762–6, Apr. 2006.
- [79] M. Wang, X. Zhou, F. Li, J. Huckins, R. W. King, and S. T. C. Wong, "Novel cell segmentation and online SVM for cell cycle phase identification in automated microscopy," *Bioinformatics (Oxford, England)*, vol. 24, no. 1, pp. 94–101, Jan. 2008.
- [80] X. Zhou, F. Li, J. Yan, and S. T. C. Wong, "A novel cell segmentation method and cell phase identification using Markov model," *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 13, no. 2, pp. 152–7, Mar. 2009.
- [81] J. Ao, S. Mitra, R. Long, B. Nutter, and S. Antani, "A hybrid watershed method for cell image segmentation," *2012 IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 29–32, Apr. 2012.

- [82] H.-H. Lin, M.-R. Tsai, C.-F. Chen, S.-Y. Chen, Y.-H. Liao, G. G. Lee, and C.-K. Sun, "Cell segmentation and NC ratio analysis of third harmonic generation virtual biopsy images based on marker-controlled gradient watershed algorithm," *2012 IEEE International Symposium on Circuits and Systems*, pp. 101–104, May 2012.
- [83] A. dos Anjos, A. L. B. Møller, B. K. Ersbøll, C. Finnie, and H. R. Shahbazkia, "New approach for segmentation and quantification of two-dimensional gel electrophoresis images.," *Bioinformatics (Oxford, England)*, vol. 27, no. 3, pp. 368–75, Feb. 2011.
- [84] S. Tse, L. Bradbury, J. W. L. Wan, H. Djambazian, R. Sladek, and T. Hudson, "A combined Watershed and Level Set Method for Segmentation of Brightfield Cell Images," vol. 7259, p. 72593G–72593G–10, Feb. 2009.
- [85] B. Albahari, P. Drayton, and B. Merrill, *C# Essentials*. O'Reilly Media, Incorporated, 2002.
- [86] A. Mahmood and S. Khan, "Correlation-Coefficient-Based Fast Template," vol. 21, no. 4, pp. 2099–2108, 2012.